GE
Security

User Manual

Picture Perfect v4 Import/Export

Picture Perfect 4.0 Import/Export

User Manual

Copyright Copyright © 2007, GE Security Inc. All rights reserved.

This document may not be copied or otherwise reproduced, in whole or in part, except as specifically permitted under US and international copyright law, without the prior written consent from GE.

Document number/460219007B (January 2007).

Disclaimer THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. GE ASSUMES NO RESPONSIBILITY FOR INACCURACIES OR OMISSIONS AND SPECIFICALLY DISCLAIMS ANY LIABILITIES, LOSSES, OR RISKS, PERSONAL OR OTHERWISE, INCURRED AS A CONSEQUENCE, DIRECTLY OR INDIRECTLY, OF THE USE OR APPLICATION OF ANY OF THE CONTENTS OF THIS DOCUMENT. FOR THE LATEST DOCUMENTATION, CONTACT YOUR LOCAL SUPPLIER OR VISIT US ONLINE AT WWW.GESECURITY.COM.

This publication may contain examples of screen captures and reports used in daily operations. Examples may include fictitious names of individuals and companies. Any similarity to names and addresses of actual businesses or persons is entirely coincidental.

Trademarks and patents GE and the GE monogram are registered trademarks of General Electric.
Picture Perfect product and logo are trademarks of GE Security.

Other trade names used in this document may be trademarks or registered trademarks of the manufacturers or vendors of the respective products.

Intended use Use this product only for the purpose it was designed for; refer to the data sheet and user documentation. For the latest product information, contact your local supplier or visit us online at `www.gesecurity.com`.

Contents

Figures vi
Preface vii
Conventions used in this document vii
Safety terms and symbols vii

Chapter 1. Installing Import/Export 1

Installation 2

Chapter 2. Functional overview 5

Import 6
Preparation for Import 6
Export 6
Preparation for Export 6
Things to Think About 7

Chapter 3. Importing data 9

Overview 10
The importdb Command 10
Options 10
Examples of importdb command 11
Database Inserts 11
Database Updates 12
Table changes: Picture Perfect versions 2.0 and 4.0 12
Import Data Files 13
Header formats 13
Example of header 17
Examples of Import Data Files 19
Illegal Characters 20
Multiple Host BID Formats 21

Chapter 4. Exporting data 23

Command Line Queries 24
selectcol 24
query 24
selectit 25
selectrpt 25
selectfile 26
SQL Queries 27
Syntax 27
Special Words and Symbols 28
Exporting History Data 28
Alarm History 28
Badge History 29
Operator History 29
Examples 30
Export Queries 30
Merging Table Rows 30
Resolving Intertable Dependencies 31
Subqueries 31

Chapter 5. Remote Procedure Call Interface 33

Overview 34
How RPC Works 34
Software requirements 35
Sources for PC-based TCP/IP Software 35
The Import/Export Interface RPC Toolkit 36
The Test Program, client 37
Application Layer Services 38
Troubleshooting 39
Communication Failures 39
Application-Related Errors 40
Verify Service 40
Verify Prototype 41
Verify Diagnostics 41
Import Service 42
Import Prototype 42
Import Diagnostics 42
Export Service 43
Export Prototype 43
Export Diagnostics 43
Running Concurrent Clients 44
Server Configuration 44
Client Configuration 44
Timeouts 45
Inventory of source code supplied with Import/Export 45

Chapter 6. Troubleshooting and support 47
Error Messages. 48
Contacting technical support 53
Online publication library 53
Appendix A. Table layouts 55
Notes 56
Alarm History (export only) 57
Person 59
Badge_History (export only) 62
Badge 66
Category 69
Facility 69
Department. 70
Host BID Format. 71
Operator History (export only) 72
Person_user 73
Person_category 74
Area 75
Area_event 77
Area_category 80
Area_event_category 81

Figures

Figure 1. Example of person record insert 19
Figure 2. Example of person record update 19
Figure 3. Example of badge record insert 19
Figure 4. Default Badge Encode Format 21
Figure 5. Sample query host_bid_format 21
Figure 6. Table Layout Example 1 56
Figure 7. Table Layout Example 2 56

Preface

This manual is intended for systems integrators and/or network programmers seeking methods for connecting to the Picture Perfect platform using asynchronous or Local Area Network (LAN) communications.

This manual provides information for a knowledgeable SQL programmer to interface external programs to the Picture Perfect database. It is not intended as a tutorial on SQL programming. If you need additional information on SQL, please contact your GE Customer Support Representative for a list of recommended sources.

Read these instructions and all ancillary documentation entirely before installing or operating this product. The most current versions of this and related documentation may be found on our website. Refer to *Online publication library* on page 53 for instructions on accessing our online publication library.

Note: A qualified service person, complying with all applicable codes, should perform all required hardware installation.

Conventions used in this document

The following conventions are used in this document:

Bold	Menu items and buttons.
Italic	Emphasis of an instruction or point; special terms.
	File names, path names, windows, panes, tabs, fields, variables, and other GUI elements.
	Titles of books and various documents.
Blue italic	(Electronic version.) Hyperlinks to cross-references, related topics, and URL addresses.
`Monospace`	Text that displays on the computer screen.
	Programming or coding sequences.

Safety terms and symbols

These terms may appear in this manual:

CAUTION: *Cautions* identify conditions or practices that may result in damage to the equipment or other property.

WARNING: *Warnings* identify conditions or practices that could result in equipment damage or serious personal injury.

Chapter 1 Installing Import/Export

This chapter covers the background information needed to successfully install your Picture Perfect Import/Export package.

In this chapter:

Installation 2

Installation

To install Picture Perfect Import Export after the base system has been installed, do the following:

1. Log in to the system as `root`.
2. From a command prompt, open a terminal window.
3. Type the following to shut down Picture Perfect:

```
. /cas/bin/profile
rc.pperf -k
```

4. Insert the Picture Perfect CD into the CD-ROM drive.

Linux

5. The CD mounts automatically.

AIX

Mount the CD by typing:

`mntCD` Enter

If this command is not found, type:

`mount -v cdrfs -o ro /dev/cd0 /mnt` Enter

6. To display a list of installation options, type:

Linux

`/media/cdrom/INSTALL -o` Enter

AIX

`/mnt/INSTALL -o` Enter

Messages similar to the following will display.

```
The following BASE_OPTIONS product(s) are available:
Prod #        Name and Descriptions
------        ---------------------
0             base             Picture Perfect Base package
1             EIF              Picture Perfect External Interface package
2             graph            Graphics Monitoring and Control package
3             image            Picture Perfect Imaging package
4             impexp           Picture Perfect Import/Export package
5             netlan           Picture Perfect Network System - Host package
6             pprs             Picture Perfect Redundant System package
7             subhost          Picture Perfect Network System - Subhost package
8             tours            Picture Perfect Guard Tours package
Enter product number(s), separated by ',' to select, 'q' to quit:
```

7. To select the Picture Perfect Import/Export package, select the appropriate number for the `impexp` product and press Enter.

   Your package selection will now be displayed, and you will be asked to confirm:

```
You have selected the following product(s):
4         impexp              Picture Perfect Import/Export package
Is this correct (y/n)? [y]
```

8. To make a different selection, type `n`, and you will be returned to *step 7* of this section, where you will be prompted again for your selection. To continue the installation, type `y`, or press Enter to accept the default (`y`).

The installation will begin, and messages similar to the following will appear on the screen:

```
Installing impexp...
Picture Perfect Multi-package Installation - 3.0 02/10/05
Copyright (C) 1991 thru 2005 GE Security
Installing from image in /mnt/cdrom/pp ...
296 blocks
Do you want to install the Picture Perfect IMPEXP Package (y/n)? [y]
```

9. Type `y` and press Enter to confirm that you want to install the Picture Perfect Import/Export package.

   You will see messages similar to the following:

```
Checking if need to save nls or help files...
Picture Perfect NLS Text Save - 3.0 1/16/05
Copyright (C) 1994-2005 GE Security
Tue Jan 21 11:53:15 EST 2005
This package has no nls or help files to save...
IMPEXP.INST Picture Perfect Installation - Version 1.6 11/15/04
Copyright (C) 1990-2002 GE Security
Loading Import/Export, Import/Export Programmer Interface from media...
2504 blocks
The source files to the Programmer Interface have been installed in the directory /
usr/rpc'
Import/Export can be configured to run multiple copies of the 'importd' server to
improve RPC response time at sites using multiple clients. For example, if you will
be using two remote badging systems and also intend to export data to a mainframe,
you can configure Import/Export to run 3 copies of the server. (Refer to the Import/
Export Guide for client implementation.)
How many instances of the server do you wish to configure?
Enter a number from 1 to 16:
```

10. Enter the number of Import/Export server copies you wish to run, then press Enter.

```
Starting the Informix database... Done.
Adding record to terminal...
Adding 1 record to tps_daemons...
Adding entry to services...
Starting the portmapper...
Shutting down the Informix database. Done.
The 'IMPEXP.INST' installation has completed successfully.
Checking if need to update nls files...
Picture Perfect NLS Check - 2.0 01/16/05
Copyright (C) 1997 = 2005 GE Security
Tue Jan 21 11:15:59 EST 2005
No nls files for impexp package
Running /cas/bin/fixperm on /tmp/impexp.perm file...
No errors detected
/cas/bin/fixperm finished.
---Updating hwdata-0.14.1-1.noarch.rpm ---
package hwdata-0.14.1-1 is already installed
INFO: Updating Red Hat patches was successful
The INSTALLation has completed.
The system needs to be rebooted for the changes to take effect.
Reboot the system (y/n)? [y]
```

Installation of the Import/Export package is complete. Type `y` to reboot your system.

Chapter 2 Functional overview

This chapter provides an overview of the functions of your Picture Perfect Import/ Export package and things you should consider when using it.

In this chapter:

Import 6
Preparation for Import 6
Export 6
Preparation for Export 6
Things to Think About 7

Import

“Import” means bringing information into the Picture Perfect database from another source, and is used to eliminate entering identical information into more than one database. For example, if personnel information such as name, address, department, identification number, and employment status is already available in a Human Resources database, that information can be imported into Picture Perfect as personnel data. Access control information can then be added to it either as it is being formatted for import, or after it has been imported.

Personnel and department information are the most common types of data imported into Picture Perfect. Import data can come from removable media or directly from another computer using a network. One or more import records can be taken from a single file.

Once the data file resides on Picture Perfect’s hard disk, import data may be submitted by the system scheduler using a **`cron`** or **`at`** job or from the Unix command line. Transaction and error status messages are posted in the output and error files given in the application header.

Preparation for Import

The imported data must be extracted from a source system, formatted to conform with the appropriate Picture Perfect record structure, and written to a file. Records to be imported must observe column positioning, with all columns represented in all records. Null fields are represented by adjacent field separators. In some records, specific columns must contain data (not null) for successful import. This requirement may depend on either the database schema or the transaction code included in the application header of the input file. The data file must then be transferred to the hard disk on the Picture Perfect host.

Typically, a system administrator imports the data from the operating system command line or sets up a scheduled event to import the data at a particular day, date, and/or time. The output and error files should be checked upon completion of the import operation.

Export

“Export” means to retrieve information (usually transaction history records) from the Picture Perfect database and place it as delimited text records in a hard disk file, making it available for transportation to another system for reporting or data synchronization purposes. Exporting does not change or remove any information from the Picture Perfect database.

Data retrieval is done through the use of command-line queries using SQL syntax.

Preparation for Export

You must define exactly what information is required on the destination system (content, order, sorting, type, etc.). The SQL query that retrieves this information must be created and thoroughly tested on the Picture Perfect system using the REPORTS form. When it meets your needs, save the SQL statement.

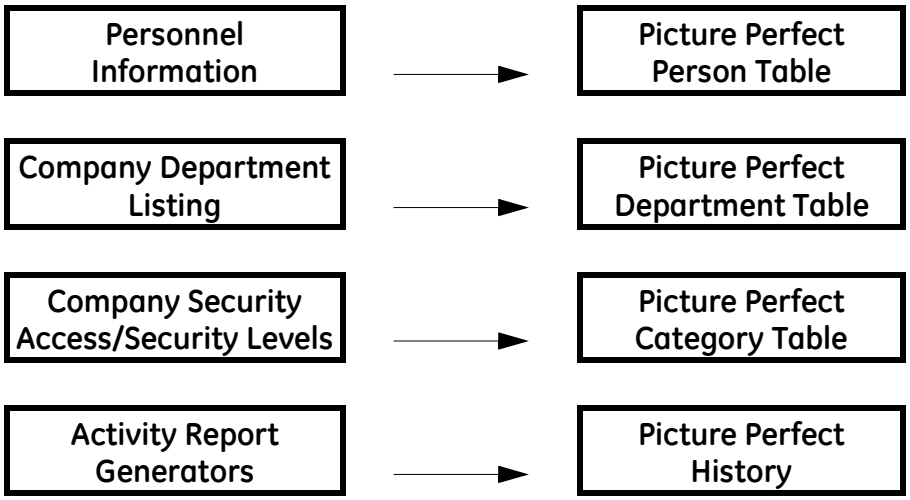
In a relational database, data is stored on the disk in areas set aside for it during installation. As installation continues, these areas are divided into tables, such as the badge table, reader table, and badge-history table. Data is stored in the tables as records, with records divided into columns. Each record in the same table represents the same type of data. Columns are positional within a record, with the same type of information stored in the same position in each record in the table. Additionally, one of the main objectives of a relational

database management system (RDBMS) is to eliminate the need for data redundancy through the use of record pointers, so that many fields (such as the category and department fields in a badge record) are integer pointers to corresponding records in other tables. This may require somewhat complex queries to retrieve enough information for the user to understand the generated report.

The design, that is the size, table name, column names, and attributes, of a relational database is referred to as its schema. Much of the Picture Perfect schema is available to the system administrator under the REPORTS interface.

Things to Think About

- Do the source and Picture Perfect databases need to stay synchronized? If a field is imported, then changed on the Picture Perfect system, the databases will not be identical (synchronized). If synchronization is required, an export is required.
- The badge encode number (the number inside the badge) is a required column in a badge record. If it is not available on the source system, how will it be assigned? It must be a real badge encode number, because it cannot be changed from the badge window after it is in the Picture Perfect database.
- Is the data that is being exported consistent with your site's security requirements and/or union rules?
- Is the system configured to route all required transactions to history? Activity not routed to history cannot be retrieved, which can distort transaction reporting on a foreign system.



Chapter 3 Importing data

This chapter describes how to import data into the Picture Perfect database, through the use of the importdb command and import files.

In this chapter:

Overview 10
The importdb Command 10
Database Inserts 11
Database Updates 12
Table changes: Picture Perfect versions 2.0 and 4.0 12
Import Data Files 13
Multiple Host BID Formats 21

Overview

Data imported into Picture Perfect is handled by the `importdb` command. In a redundant Picture Perfect system, data may be imported only to the primary host.

Imported data records can perform either an add or a change to the Picture Perfect database, depending on the transaction type in the application header. When `importdb` is used to add or change records, updates are downloaded to the appropriate microcontrollers just as if the changes were made through Picture Perfect itself. The import operation may involve one or more records in a single file.

Import data may be submitted by the system scheduler using a `cron` or `at` job or from the Unix command line. The login ID assigned to the job must be that of a Picture Perfect operator with permission to access the tables involved in the operation. The login ID must also have read and execute rights to the directory structure for the input, output, and error file paths; read permission for the input file itself; and write permission to the output and error files and their parent directories.

An import file is made up of three parts: a Presentation header, an Application header, and data records. The import file begins with a two-part header. The first eight bytes make up the presentation header, which defines the ASCII data type and carriage control (new lines, carriage returns, or both). The next 120 bytes make up the application header, which defines the Picture Perfect computer's network hostname, the import owner's login name and password, the import transaction type, the affected table, the field and record delimiters, the output and error file names, two user-comment fields, and the import signature. There is only one header per file, although the file may have many import records. There are examples of header files in the `/cas/import` directory. See *Header formats* on page 13 for more information on headers.

The importdb Command

Syntax: `importdb -i filename [-o filename] [-e filename]`

Usage: `importdb -i input_file` is a required argument

`[-o output_file ]` (default: use header record)
`[-e error_file ]` (default: use header record)
`[-m ]` (use `modify_date` and `modify_time` supplied in input file)

Options

`-i filename`

Input File

`filename` is the file containing the data for import, including headers. If `filename` is not a full pathname (denoted by a leading slash, "/"), the file is assumed to be located in the `/cas/import` directory.

`-o filename`

Output File

`filename` receives the redirection of standard output (`stdout`). If `filename` is not a full pathname (denoted by a leading slash, "/"), the file is written to the `/cas/export` directory. This command line option has precedence over the application header output file field. The application header determines whether the output file is overwritten or appended.

`-e filename`

Error File

`filename` receives the redirection of standard error (`stderr`). If `filename` is not a full pathname (denoted by a leading slash, "/"), the file is written to the `/cas/export` directory. This command line option has precedence over the application header error file field. The application header determines whether the error file is overwritten or appended.

Examples of importdb command

`importdb -i badge.dat`

In this example, `importdb` references the file `/cas/import/badge.dat` for headers and import data; output and error files are taken from the application header.

`importdb -i /tmp/badge.dat`

In this example, `importdb` references the file `/tmp/badge.dat` for headers and import data; output and error files are taken from the application header.

`importdb -i badge.dat -o out.stat`

In this example, `importdb` references the file `/cas/import/badge.dat` for headers and import data and writes standard output (`stdout`) messages into `/cas/export/out.stat`. Error file names are taken from the application header.

`importdb -i /tmp/badge.dat -o /tmp/out.stat`

In this example, `importdb` references the file `/tmp/badge.dat` for headers and import data and writes standard output (`stdout`) messages into `/tmp/out.stat`. Error file names are taken from the application header.

`importdb -i badge.dat -o out.dat -e err.dat`

In this example, `importdb` references the file `/cas/import/badge.dat` for headers and import data; and writes standard output (`stdout`) to the file `/cas/export/out.dat`. It writes error (`stderr`) messages into the file `/cas/export/err.dat`.

`importdb -i /u/imp/badge.dat -o /u/exp/out.dat -e /tmp/err.dat`

In this example, `importdb` references the file `/u/imp/badge.dat` for headers and import data; and writes standard output (`stdout`) to the file `/u/exp/out.dat`. It writes error (`stderr`) messages into `/tmp/err.dat`.

Database Inserts

Adds (inserts) to any table must conform to the following requirements:

- The **Modify Date** and **Modify Time** fields should always be completed. Use the current date (YYYYMMDD) and current time (HHMMSS). If these fields are not filled in, the current date and time on the server will be used.
- Refer to *Header formats* on page 13 for Person table options regarding use of either **Department ID** or **Department Description**.
- Refer to *Appendix A, Table layouts* for any required fields for the particular table to which you are adding records. Be sure to read the notes on page .56.

- When adding personnel records, make sure the departments referenced already exist in the Picture Perfect database. (If not, add the required departments to the **Department** table before adding the personnel records.) If the departments do not exist when the personnel records are added, an error will result.
- Adds to a Picture Perfect Enterprise system are handled as follows:
  - Any inserts into the **Badge** or **Person** table on a subhost are inserted into the network-host **Badge** or **Person** table first. Then, upon a successful insert into the network-host **Person** table, if there is any user or category data on the personnel record, the associated `person_user` and `person_category` records are first created on the network host, then on the local subhost. If the `person_user` and `person_category` records are inserted successfully, the personnel record is inserted into the local subhost with the same ID obtained from the network host. If the network is down, a badge or person insert will not be allowed.
  - Import/Export prevents insertion of a category, department, or permission-group record on a subhost. Since these tables are global in a network, they can only be inserted on the network host.

Database Updates

Changes (updates) to any table are handled as follows:

- Any column that contains data is updated in the Picture Perfect database except for the ID or Badge ID (bid) columns. For security reasons, these columns cannot be updated through the import program.
- If you do not want to update the data of a particular column, do not place any data between the column delimiters. The current data will remain intact.
- To remove data from a column in the Picture Perfect database, insert the word **NULL** in that column. The current information will be erased. This applies to both integer and character data types.
- For changes to the Badge Table, the **id** and **bid** fields are required in order to automatically update all the micros where the badge resides. If the **bid** is not provided, then an attempt will be made to acquire it using the **id** field, if present.
- After the changes are made, any column that resides on the microcontroller is automatically updated if the data is different from the data already existing in the database.

Table changes: Picture Perfect versions 2.0 and 4.0

Beginning in Picture Perfect 2.0, the area table was split, and the category data was moved to a separate tables. Two new tables now contain that data: `area_category` and `area_event_category`. In Picture Perfect 2.0, categories were increased from 10 to 32.

For all of the new 'sub-tables', the foreign key back to the 'parent' table is the table id of the parent table, that is:

- The `area_category.area_id` field is a foreign key to the `area.id` field in the `area` table.
- The `area_event_cat.area_event_id` field is a foreign key to the `area_event.id` field in the `area_event` table.

Also beginning in Picture Perfect 2.0, the `badge` table was split, and the category and user data was moved to separate tables. In Picture Perfect 4.0, the `badge` table was further split into a `badge` and `person` table. Categories that can be assigned to personnel records were increased to 96. For the `person` table, the

categories were moved to the `person_category` table, and the user data was moved to the `person_user` table.

For all of the new 'sub-tables', the foreign key back to the 'parent' table is the table id of the parent table, that is:

- The `person_user.person_id` field is a foreign key to the `person.id` field in the `person` table.
- The `badge.person_id` field is a foreign key to the `person.id` field in the `person` table.
- The `person_category.person_id` field is a foreign key to the `person.id` field in the `person` table.

As a result of the above-mentioned changes, an insert into one of the above tables that included data in one or more of the sub-tables, would have resulted in 3 (area or `area_event`) or 4 (`person`) additional steps. An insert into the person table, for example, that included badge, user and category data, would have involved:

1. Inserting the basic personnel record.
2. Exporting it to get back the `person.id` of the inserted record.
3. Inserting the badge records using the exported `person.id` for the `person_id` field.
4. Inserting the person_category records using the exported `person.id` for the `person_id` field.
5. Inserting the person_user records using the exported `person.id` for the `person_id` field.

In order to simplify the import/update of records into these tables, the Import/Export program was modified. When inserting records into these tables, the extra columns for the sub-tables must be specified. The Import/Export program will handle the creation of the sub-table records into their appropriate tables.

See *Examples of Import Data Files* on page 19.

Import Data Files

The import data file contains the records to be imported into the database.

Header formats

Use the Presentation Header and Application Header formats when creating import files. There are examples of header files in the `/cas/import` directory.

Presentation Header

The first eight bytes in the input file make up the presentation header. Important points about the presentation header are:

- Fields are positionally dependent. Each field must start on the exact character listed below for a file to import successfully.
- Fields may be less than (but never more than) the indicated number of characters. If the content of the field doesn’t fill the entire field, fill the remaining character positions with spaces. Use of tabs is discouraged, as they may be interpreted either as a single character or as multiple spaces and could, therefore, jeopardize the success of the import.
- Headers are case sensitive. Keep upper-case and lower-case characters consistent and inviolate.

Byte 0 Data Coding Format (1 byte)

A ASCII

Byte 1 Carriage Control for the Import File (1 byte)

The type of line ending used by the exporting system:

0 New lines only.

1 Carriage returns only.

2 Carriage returns followed by new lines.

Bytes 2-7 Reserved (6 bytes)

Must be all zeros (“000000”).

Application Header

After the eight-byte presentation header, the next 120 bytes make up the application header. Important points about the application header are:

- Fields are positionally dependent. Each field must start on the exact character listed below for a file to import successfully.
- Fields may be less than (but never more than) the indicated number of characters. If the content of the field doesn’t fill the entire field, fill the remaining character positions with spaces. Use of tabs is discouraged, as they may be interpreted either as a single character or as multiple spaces and could, therefore, jeopardize the success of the import.
- Headers are case sensitive. Keep upper-case and lower-case characters consistent and inviolate.

Bytes 8-23 Picture Perfect Computer Network Hostname (16 bytes)

The internet hostname of the Picture Perfect system where the database resides on disk. Configured in the `/etc/hosts` file; can be displayed from the command line using the `hostname` command.

Bytes 24-31 Picture Perfect Operator Login Name (8 bytes)

A valid Picture Perfect operator login name with a Permission that allows access to the table being modified by the import; write and execute access to the input, output, and error file directories; and write access to the input, output, and error files themselves.

Bytes 32-39 Reserved (8 bytes)

Must be spaces.

Bytes 40-41 Transaction Code (2 bytes)

Controls the import function.

31 Update existing records in any table with an operator-activity transaction for each modification.

- Each input data record must have the employee number of a valid Picture Perfect operator added as an extra field at the end of the record.
- If the table is not the `badge` table, the `id` field must never be blank.
- If the table is the `badge` table, either the `id`, or the `bid` must be included.

32 Update existing records in any table with one operator activity transaction for the entire import function.

- If the table is not the `badge` table, the `id` field must never be blank.
- If the table is the `badge` table, either the `id,` or the `bid` must be included.

33 Insert new records in any table.

- Badge encode number must be included for badge records.
- Import creates record id.

34 Insert new badge record(s).

- Badge encode number must be included for badge records.
- Import creates record id.
- Application-Header byte 123 controls the content of the dept column.

35 Insert new record(s) in any table.

- Import records must include record ids.

41 Insert new record(s) to table specified in the application header.

- The `id` field must be blank and the `description` field must never be blank.
- If the table is the badge table, the `id` field must be blank and the `bid` field must never be blank.
- Logging of Operator History will be based on the `login_id` field obtained from the contents of Application-Header bytes 24-31 (Picture Perfect operator login name).

42 Update existing record(s) of table specified in the application header.

- If the table is not the `badge` table, the `id` field must never be blank.
- If the table is the `badge` table, either the `id`, or the `bid` must be included.
- Logging of Operator History will be based on the `login_id` field obtained from the contents of Application-Header bytes 24-31 (Picture Perfect operator login name).

Bytes 42-59 Picture Perfect Table Name (18 bytes)

Name of the Picture Perfect table to be modified by the import; if the transaction code is 30 or 34, the table name must be `badge`.

Byte 60 Column Delimiter Character (1 byte)

The character, usually the vertical bar (|), that separates columns in the import records.

Byte 61 Record Delimiter Character (1 byte)

The character, usually a new line, that separates import records in the file.

Bytes 62-75 Output File Name (14 bytes)

The name of the file where `importdb` writes the import informational messages; if a full path name is not provided, the file will be written into `/cas/export;` if the `-o` option is used on the `importdb` command line, this field is ignored.

Byte 76 Output File Append or Overwrite (1 byte)

A Append.

- If the file doesn't exist, create and write.
- If the file does exist, add import informational messages to the end of the file.

N New.

- Any character other than A.
- If the file doesn't exist, create and write.
- If the file does exist, overwrite and destroy any previous content.

Bytes 77-90 Error File Name (14 bytes)

The name of the file where `importdb` writes error messages generated during the attempted import function. If a full path name is not provided, the file will be written into `/cas/export`. If the `-e` option is used on the `importdb` command line, this field is ignored.

Byte 91 Error File Append or Overwrite (1 byte)

A Append.

- If the file doesn't exist, create and write.
- If the file does exist, add error information to the end of the file.

N New.

- Any character other than A.
- If the file doesn't exist, create and write.
- If the file does exist, overwrite and destroy any previous content.

Bytes 92-106 Application Control Field 1 (15 bytes)

User-defined field; content not used by `importdb`.

Bytes 107-121 Application Control Field 2 (15 bytes)

User-defined field; content not used by `importdb`.

Byte 122 Reserved (1 byte)

Must be a space.

Byte 123 Person Department Control (1 byte)

I Integer in department column.

- If the department column in a person record is not NULL, it must be an integer pointer to a valid record in the department table.

D Description in department column.

- If the `department` column in a `person` record is not NULL, it must contain the full description of a valid department in the `department` table.
- Any character other than **I** causes this action on a `person` import.
- Unused for imports to other tables.

Byte 124 Badge Update Control (1 byte)

A Update any badge record(s) as requested, regardless of status.

V Update only badge records with a valid status.

- Any character other than A causes this action on a badge import.
- Unused for imports to other tables.

Byte 125-126 Picture Perfect Export Signature (2 bytes)

Must be CR.

Byte 127 Header Section Terminator (1 byte)

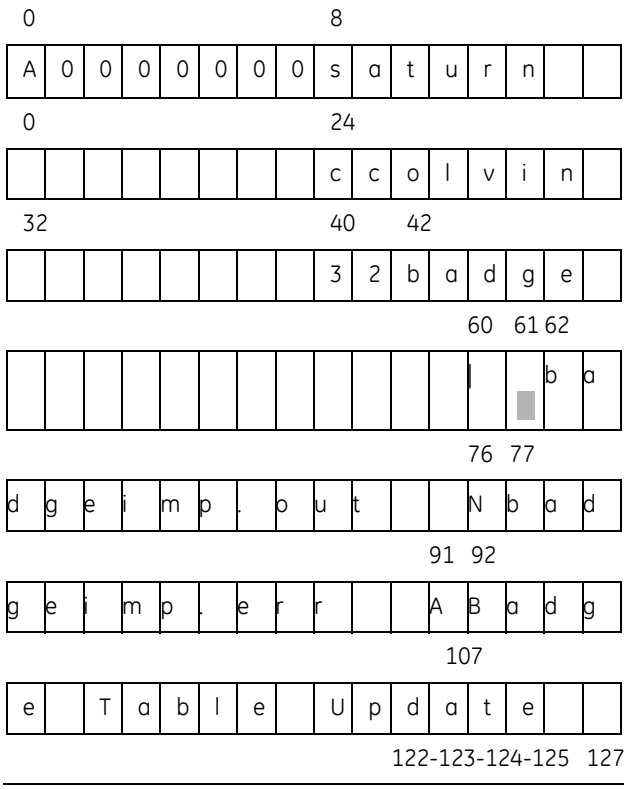
Must be a carriage return (ASCII `0x0A`).

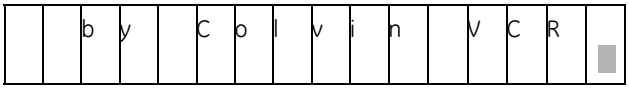
Example of header

Requirement:

Update existing badge records with valid badge status only, which includes badge encode numbers in each input record and carriage returns only to the Picture Perfect host `saturn, login name ccolvin,` overwrite output file `/cas/export/badgeimp.out`, append to error file `/cas/export/badgeimp.err,` one operator activity record for the entire function, separate the columns with a vertical bar (|), and the records with a new line ( ).

```
A0000000     saturn          ccolvin         32badge        |
badgeimp.out     Nbadgeimp.err  ABadge Table Update  by Colvin VCR
```





A	ASCII input data.
0000000	New lines.
saturn	Computer’s network hostname.
ccolvin	Operator to own the import.
32	Transaction code (update records in table).
badge	Name of Picture Perfect table to be modified.
|	Column delimiter (vertical bar).
▮	Record delimiter (new line).
badgeimp.out	Output data filename in `/cas/export`.
N	Overwrite the output file.
badgeimp.err	Error file name in `/cas/export`.
A	Append to the error file.
Badge Table Update by Colvin	User comment fields 1 and 2.
V	Update valid badges only.
CR	Import signature.
▮	Header terminator (carriage return)

Examples of Import Data Files

Person and Badge record inserts and updates

Employee information is stored in several different tables in the Picture Perfect database. These include `person, badge, person_user`, and `person_category`. Import/Export allows you to create a single record containing all relevant data for an employee, and the system will automatically insert the appropriate information into the various tables, as needed. For example, last name and first name data will be stored in the `person` table, while category ids will be stored in the `person_category` table.

The badge table in Picture Perfect 2.0 was split into the `badge` and `person` tables in Picture Perfect 4.0. So, the person record MUST include placeholders for the 31 fields of a badge record.

The file, `sample.imp`, in the `/cas/import` directory specifies the format required to import a complete person record along with user fields, categories, and badge data.

Note: You can only attach one badge record to a person record when using Import/Export.

Example 1: Person record insert

The example below shows a simple insert of a person record. The first 36 slots are the person table data, the next 40 slots are the user field data, the next 96 slots are the category data and the last 31 slots are the badge data. Make sure you have a total of 203 pipe delimiter symbols for each record to be imported

Figure 1. Example of person record insert

```
A0000000yourpphost      install          33person           |
person.out              nperson.err      n                  12              CR
||0|1||||7334||Joe|Smith|JS||25|||||||0|||||||||||-1|20041213|220905|User field 1|||||
|||||||||||||||||||||||||||||||||||||||||||||||||User field 40|-1|||||||||||||||||||||||||||||||||||||||
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||-1||Test Badge 5|2233433349|0||
||||||20041215|181322|0|||||||-1||0|1|||||-1|20041213|231322|
```

Example 2: Person record update

The example below shows an update of a person record. The first 36 slots are the person table data, the next slot is the employee number used for a transaction code type of 31 (0004 in this case), the next 40 slots are the user field data, the next 96 slots are the category data and the last 31 slots are the badge data. In this case, there is no badge being updated on the person record.

Figure 2. Example of person record update

```
A0000000yourpphost      install          31person           |
person.out              nperson.err      n                  12              CR
4||0|1||||4634||James|Smith|JS||23|||||||0|||||||||||-1|20041213|220905|0004|User field 1||||
|||||||||||||||||||||||||||||||||||||||||||||||||User field 40|1|||||||||||||||||||||||||||||||||||||||
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||96|||||||||||||||||||||||||||||||
```

Example 3: Badge record insert

Badge records can be inserted into the badge table without other employee/person data. The following sample import file contains a header followed by a badge record insert. This example shows the minimum required fields on the badge record.

Figure 3. Example of badge record insert

```
      A1           3.99.999.39   install         34badge       |
      BdgIns.out   NbadgIns.err  N                             ACR
      |Test Badge 4|2233433346|0|||||||||||||||||||||||||||-1|||
```

Illegal Characters

The Informix database manager does not allow certain characters in data fields. Make sure you filter out the following characters. For names containing an apostrophe (') such as O'MALLEY, you may replace the apostrophe with a back quote. The back quote is on the key with the tilde (~). Or you may leave a space, for example, O MALLEY.

'	single quote or apostrophe
“	double quote
*	asterisk
<	less than
>	greater than
!	exclamation point
=	equal
&	ampersand
|	pipe
^	carat

Multiple Host BID Formats

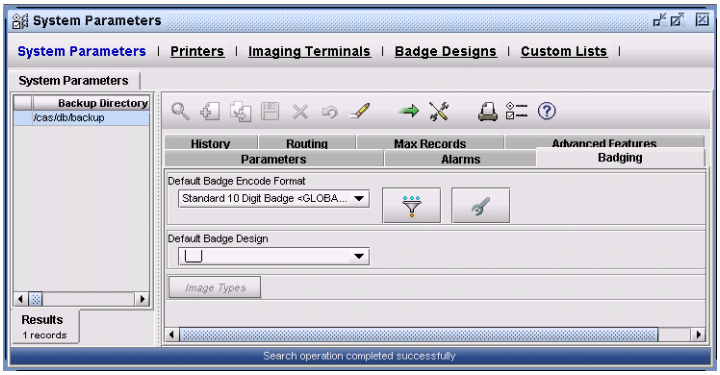
If you have multiple badge formats, and the badge records you are importing are of a format that is not the default, you must enter the ID corresponding to that badge format when creating the input record.

To determine your default badge format:

From the Picture Perfect primary navigation menu, select **Setup**, then **Parameters,** then click the **Badging** tab.

The current default badge format will be displayed.

Figure 4. Default Badge Encode Format



To import badge records that are of a different format than the default:

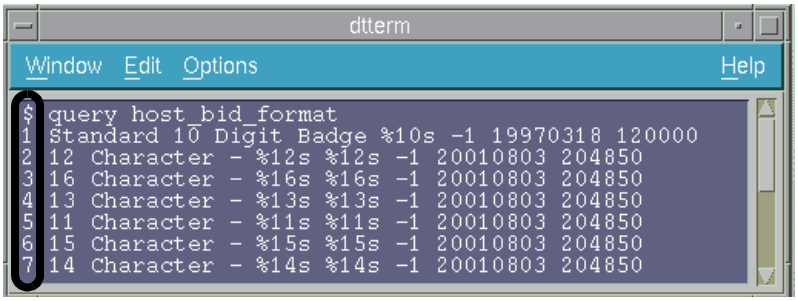
1. From a command prompt, type:

   `query host_bid_format`

   A screen similar to the following will display.

Figure 5. Sample query host_bid_format

This column represents the ID corresponding to the host BID format



2. Make a note of the ID corresponding to the host BID format you will be importing. This number must be entered as the `bid_format_id` field when inserting badges.

Chapter 4 Exporting data

This chapter describes how to export data, through the use of command line queries using SQL syntax.

In this chapter:

Command Line Queries 24
SQL Queries 27
Exporting History Data 28
Examples 30

Command Line Queries

There are several selection commands available for exporting data.

Picture Perfect selection commands reside in `/cas/bin`. If your `$PATH` environment variable does not contain the `/cas/bin` directory, it is necessary to add `/cas/bin` to your `$PATH` or prepend each of the commands with `/cas/bin/`.

When displaying information one screen at a time, follow the command below with a broken vertical bar (pipe sign: |) and the word `more`, as shown in the following example:

```
query badge | more
```

Press Spacebar for the next screen, Enter for one more line, and `Q` to quit early and return to the command-line prompt.

Note: String (or character) values must be surrounded by quotes. The quotes surrounding strings must be different from the type used to envelop the SELECT statement. For example, since single quotes are used to envelop the SELECT statement below, double quotes must surround the like CAFE% clause:

```
selectrpt 'select id, description from area where description like "CAFE%"'
```

This rule applies to all of the commands discussed in this manual.

selectcol

Syntax:

```
selectcol table_name
```

Description:

Displays the column name and type of each field in the `table_name` to the terminal only.

Example:

```
selectcol category
```

An example of the output from the above command would be:

```
 column_name                type
 --------------------       -----------
1. id                       integer
2. description              char(31)
3. permission_grp           integer
4. m2mr_type                integer
5. facility                 integer
6. modify_date              integer
7. modify_time              integer
```

query

Syntax:

```
query table_name
```

Description:

Displays all rows and all columns from `table` to the screen in an undelimited format.

Example:

```
query operator | more
```

An example of the output from the above command would be:

```
1 Systems Administrator 0000 install 1 1 -1 9e05e6832caffca519722b
608570b8ff4935b94d 0 -1 20041006 170806
-1 System System system 1 1 -1 0 -1 20041018 194738
```

selectit

Syntax:

```
selectit 'SELECT statement'
```

Description:

Displays results of the SQL Select Statement to the screen in undelimited format. This command's output is similar to `query`, except that the output is limited to those rows matching the selection criteria and those columns selected. Note that the `SELECT` statement must be enclosed in matching single or double quotes (**'**) or (**"**) in order for the shell to interpret the statement as a single parameter.

Example:

```
selectit 'select id from badge where modify_date > 20020201'
```

An example of the output from the above command would be:

```
851
852
853
854
```

selectrpt

Syntax:

```
selectrpt 'SELECT statement'
```

Description:

Displays results of SQL Select Statement to the screen, one column per line, with column names followed by corresponding data.

Example:

```
selectrpt 'select id, description from area where description like "CAFE%"'
```

An example of the output produced by the above command might be:

```
id : 1
description : CAFETERIA - MAIN DINING ROOM
id : 2
description : CAFETERIA - HIGH SECURITY
id : 3
description : CAFETERIA - LASER LAB
id : 4
description : CAFETERIA - OIL RECOVERY RESEARCH
id : 5
description : CAFETERIA - HYDROCARBONIC LAB
```

selectfile

Syntax:

```
selectfile 'SQL statement' file
```

Description:

Writes results of SQL Select Statement with delimiters to the disk file `file`, one record per line. Lines exceeding 80 characters in length will wrap to the next line.

Example:

```
selectfile 'select * from person where id < 9' person.dat
```

There will be no visible output for this command unless an error occurs. An example of the output from the above command might be (contents of `person.dat`):

```
selectfile 'select * from person where id < 4' person.dat
1||0|1||||1234||John|Smith|JS| | | | | | | | ||0|2|20061113|150632|-1|20061110|1
4359|0||||0|-1|20061114|194827|
2||0|3||||4567||Johnny|Smitka|JS| | | | | | | | ||0|2|20061113|150632|-1|2006121
|155351|0|||||-1|20061212|205442|
3||0|1||||7890||John|Rice|JR| | | | | | | | ||0|2|20061113|150632|-1|20061212|15
626|0|||||-1|20061212|205648|
```

SQL Queries

Syntax

Since SQL keywords are case-sensitive, for clarity in the following statements, the SQL keywords are given in upper case and everything else in lower case. Also for clarity, example statements are broken into multiple lines, since SQL is unconcerned with carriage control.

The format of a basic SQL statement is:

```
SELECT columns FROM tables WHERE criteria ORDER BY columns
```

columns

These are the names of columns within `tables`. Multiple column names are separated by commas. When using multiple tables, column names should include the table name (`table.column`) to avoid ambiguity and a failed export. An asterisk (**`*`**) following the `SELECT` verb indicates that all columns are to be retrieved for selected rows. Columns are written to the output file in the left-to-right order listed after the `SELECT` verb.

tables

These may be one or more Picture Perfect tables. Multiple table names are separated by commas. Any table referenced in any clause in the statement must be listed.

criteria

These specify the conditions the rows must meet to be selected. `criteria` may be tested for equality or inequality, with multiple tests connected through the use of logical conjunctions, such as `AND` or `OR`.

ORDER BY

Within the `ORDER BY` clause, rows are sorted by the given columns. The first column listed serves as the primary sort field, and additional columns sort rows where the primary sort field is identical. The default sorting order is ascending. To sort in descending order, use the keyword `DESC` after the column name in the `ORDER BY` clause.

Special Words and Symbols

Table 1. Wildcard Selection Criteria

KEYWORD	SYMBOL	MEANING
LIKE	%	Multiple characters.
	_ (underscore)	Single character.
MATCHES	*	Multiple characters.
	?	Single character.
IN	( )	Compare something to any one of the values within the parentheses.
BETWEEN x and y		Any valid value in the contiguous range, x through y, inclusive.

Table 2. Equality Comparison Operators

SYMBOL	MEANING
=	Equal to. Character strings must be defined using either quotations (“string”) or apostrophes (‘string’).
>	Greater than.
>=	Greater than or equal to.
<	Less than.
<=	Less than or equal to.
!	Not.
IS NULL	Column has no data; not the same as a value of zero.
IS NOT NULL	Column has data entered.

Exporting History Data

Three types of history transactions can be exported from the Picture Perfect system. These are Alarm History, Badge History, and Operator History.

Alarm History

Picture Perfect maintains two alarm history tables: `alarm_history` and `alarm_historytmp`. Alarm transactions are inserted into the `alarm_history` table until it reaches the `alm_hissize` parameter located in the `system_config` table (accessed through the System Parameters form).

When the number of alarm history transactions reaches the threshold or percentage of the `alm_hissize`, the `alarm_history` table is renamed to `alarm_historytmp` and the system pops up a window telling the operator to archive the alarm history data. The system then drops the current `alarm_history` table and recreates it as an empty table. New transactions are then inserted into the empty `alarm_history` table while the older transactions reside in the `alarm_historytmp` table until overwritten by the next rollover of alarm history.

When exporting alarm history data, you may select data from either table. The most recent history data is in the alarm_history table. The older alarm history is in alarm_historytmp. Both have the same table layout.

Badge History

Picture Perfect maintains two badge history tables: badge_history and badge_historytmp. Badge transactions are inserted into the badge_history table until it reaches the bdg_hissize parameter located in the system_config table (accessed through the System Parameters form).

When the number of badge history transactions reaches the threshold or percentage of the bdg_hissize, the badge_history table is renamed to badge_historytmp and the system displays a window telling the operator to archive the badge history data. The system then drops the current badge_history table and recreates it as an empty table. New transactions are then inserted into the empty badge_history table while the older transactions reside in the badge_historytmp table until overwritten by the next rollover of badge history.

When exporting badge history data, you may select data from either table. The most recent history data is in the badge_history table. The older badge history is in badge_historytmp. Both have the same table layout.

For tracking time and attendance information, use one of the transaction date (xxx_xact_date) and transaction time (xxx_xact_time) columns from the badge history table.

Operator History

Picture Perfect maintains two operator history tables: operator_his and operator_histmp. Operator transactions are inserted into the operator_his table until it reaches the opr_hissize parameter located in the system_config table (accessed through the System Parameters form).

When the number of operator history transactions reaches the threshold or percentage of the opr_hissize, the operator_his table is renamed to operator_histmp and the system displays a window telling the operator to archive the operator history data. The system then drops the current operator_his table and recreates it as an empty table. New transactions are then inserted into the empty operator_his table while the older transactions reside in the operator_histmp table until overwritten by the next rollover of operator history.

When exporting operator history data, you may select data from either table. The most recent history data is in the operator_his table. The older operator history is in operator_histmp. Both have the same table layout.

Examples

Export Queries

Requirement:

Retrieve the last name, badge encode number, reader description, transaction date, and transaction time that occurred between January 5 and January 7, 2005, sorted by last name, using only the current history table.

Statement:

```
SELECT last_name,bid,reader_desc, host_xact_date,host_xact_time
FROM badge_history
WHERE host_xact_date BETWEEN 20050105 AND 20050107
ORDER BY last_name
```

Requirement:

Retrieve the last name, badge encode number, and reader description of all badges with “00124” as the first 5 digits of the badge encode number that were used between January 1 and January 5, 2005, in the current history table, sorted descending by badge encode number.

Statement:

```
SELECT last_name,bid,reader_desc
FROM badge_history
WHERE bid LIKE '00124%'
AND host_xact_date BETWEEN 20050101 AND 20050105
ORDER BY bid DESC
```

Merging Table Rows

Columns listed with the SELECT verb are reported left to right in the order listed. When rows from multiple tables need to be merged, the UNION keyword is used to connect two separate selections. The sorting column is given positionally, with the ORDER BY clause given only after the last statement.

Requirement:

Retrieve the last name, badge encode number, and reader description of all badges with “00124” as the first 5 digits of the badge encode number that were used January 1 through 5, 2005, sorted by last name, merging rows from the current and rollover badge history tables.

Statement:

```
SELECT last_name,bid,reader_desc
FROM badge_history
WHERE bid LIKE '00124%'
AND host_xact_date BETWEEN 20050101 AND 20050105
UNION
SELECT last_name,bid,reader_desc
FROM badge_historytmp
WHERE bid LIKE '00124%'
AND host_xact_date BETWEEN 20050101 AND 20050105
ORDER BY 1
```

Resolving Intertable Dependencies

Many columns contain pointers into other tables to conserve space and minimize data redundancy. Pointers indicate the record identification number of the record in a referenced table. To retrieve the record description rather than the pointer, compare the pointer value to the record ID in the referenced table, as in the following example. Also note the use of the qualified column names (`badge_history.area` and `area.id`) to remove all doubt as to where the column resides (`badge_history` or `area`).

Requirement:

> Retrieve the last name, badge encode number, area, and reader descriptions of all badges with “00124” as the first 5 digits of the badge encode number that were used January 1 through 5, 2005, sorted by last name, merging rows from the current and rollover badge history tables.

Statement:

```
SELECT last_name,bid,area.description,reader_desc,
FROM badge_history,area
WHERE bid LIKE '00124%'
AND badge_history.area = area.id
AND host_xact_date BETWEEN 20050101 AND 20050105
UNION
SELECT last_name,bid,area.description,reader_desc
FROM badge_historytmp
WHERE bid LIKE '00124%'
AND badge_history.area=area.id
AND host_xact_date BETWEEN 20050101 AND 20050105
ORDER BY 1
```

Subqueries

A selection criterion can depend on the result of another `select` statement. This requires the use of a subquery, the result of which is passed back as one or more conditions to be met. The inner query functions as the subquery, and is resolved first. It may return one or more values of the data type needed as criteria, as in the following example.

Requirement:

> Retrieve the last name, badge encode number, reader and area descriptions, and transaction dates and times for all badge activity by badgeholders in the SECURITY department, sorted by date and time.

Statement:

```
SELECT last_name,bid,area.description,
reader_desc,host_xact_date,host_xact_time
FROM badge_history,area
WHERE badge_history.dept = (SELECT id FROM department
WHERE description='SECURITY')
UNION
SELECT last_name,bid,area.description,
reader_desc,host_xact_date,host_xact_time
FROM badge_historytmp,area
WHERE badge_history.dept = (SELECT id FROM department
WHERE description='SECURITY')
ORDER BY host_xact_date,host_xact_time
```

Chapter 5 Remote Procedure Call Interface

This chapter describes the Remote Procedure Call (RPC) Interface which provides remote hosts or workstations with real-time access to the Picture Perfect database, eliminating the need to log in to the Picture Perfect host to execute command-line queries or run the `importdb` program manually.

In this chapter:

Overview 34
How RPC Works 34
Software requirements 35
The Test Program, client 37
Application Layer Services 38
Troubleshooting 39
Verify Service 40
Import Service 42
Export Service 43
Running Concurrent Clients 44
Timeouts 45
Inventory of source code supplied with Import/Export 45

Overview

The Remote Procedure Call (RPC) Interface provides remote hosts or workstations with real-time access to the Picture Perfect database, eliminating the need to log in to the Picture Perfect host to run the `importdb` program manually. By compiling and linking the supplied routines with the client application, import and export functions can be performed on the Picture Perfect host as though they were executing locally.

Three high-level functions (or services) are supported:

- Verify existence of Picture Perfect database server.
- Import records into Picture Perfect database.
- Export records from Picture Perfect database.

When one of the above services is requested, the results are almost immediately available in the form of a local data structure within the client's address space. Delay time is generally proportional to the amount of data being exchanged and the type of service the caller is requesting.

Note: The function, "bidgen" is no longer supported.

How RPC Works

An RPC client requests a known service from the server by creating a connection to the RPC server on the specified host for the duration of the RPC. Data is then sent to the server by passing it through a set of XDR filters that convert it to a format understood by all types of computers. These filters pass the converted data through the connection and across the network to the specified RPC server. The server converts the data to its own machine's data format and performs the requested service. The results are sent back to the client in the same fashion by passing the returned data through the same set of XDR filters. These XDR routines are included with the client RPC libraries.

Software requirements

In order to write to the RPC Interface, the following items are required:

- RPC Toolkit for Picture Perfect (included with Import/Export software).
- Picture Perfect Import/Export User Manual (this manual).
- Third-party RPC and XDR Library.
- Third-party TCP/IP Socket Library.
- Third-party TCP/IP Runtime Services.
- Third-party C Language compiler and linker.

When running the RPC interface with Picture Perfect 4.0, the following customer environments are supported:

Table 3. Supported systems

Operating System	Compiler	TCP/IP Library
MS Windows XP Professional	MSVC++ 4.X	ONC/RPC
MS Windows 2000	MSVC++ 4.X	ONC/RPC
Red Hat Linux 4.0 (AS, ES, or WS)	GNU gcc	RedHat, Inc.
Red Hat Linux 3.0 (AS, ES, or WS)	GNU gcc	RedHat, Inc.
IBM AIX 5L version 5.3 for RISC/6000	IBM C	IBM AIX 5.3
IBM AIX 5L version 5.2 for RISC/6000	IBM C	IBM AIX 5.3
MVS	MVS	MVS

Sources for PC-based TCP/IP Software

The companies listed below are suppliers of TCP/IP development and runtime environments for DOS/ Windows environments.

Netmanage, Inc.
10725 North De Anza Boulevard
Cupertino, CA 95014 - 2030
Phone: (408) 973-7171
http://www.netmanage.com

Distinct Corporation
3315 Almaden Expressway, Ste. 10
San Jose, CA 95118
Phone: (408) 445-3270
Copyright Martin F. Gergeleit
http://www.onc-rpc-xdr.com

The Import/Export Interface RPC Toolkit

The toolkit consists of seven C Language source files, a header file, and a make file. The C source files, when compiled and linked with the RPC libraries supplied by one of the above vendors, create a demonstration client program called client. This can be used as a boilerplate for developing the production version of the intended client application. The toolkit is delivered in source-code form rather than relocatable object form so the software can be run on a wide variety of machines and developed with a wide range of compilers, linkers, and libraries. Although the standard RPC and XDR routines are 99 percent compatible across vendors' libraries, there could be nuances specific to a given vendor's library that would make it impossible to compile and link to relocatable object files without the ability to make minor modifications.

The client application programmer should begin by getting the demonstration program to compile and link. Very few changes to the source code should be needed to accomplish this. The program should then be tested with a Picture Perfect host upon which the Import/Export package is installed. After the program is functional and all of the services are verified, it is suggested that the programmer make a copy of the working version of code and replace the top level functions supplied with the demonstration program with his or her own customized functions designed to be called from within the customer's application.

After the Import/Export package has been installed, the toolkit will be found in the directory `/usr/rpc`. A makefile must be written for DOS/Windows environments. It is necessary to set the following flags and pass them to the compiler, depending upon the environment:

Table 4. Set flags

OS type	# define
Microsoft C	# define _MSDOS
AIX 5L	# define _AIX52 # define _AIX53
Linux	# define LINUX
VM	# define VM
MVS	# define MVS

The Test Program, client

This program, when operational, exercises each of the three services. Its usage is shown below:

(notice that `-h host` is required with all variations of the program's usage.)

```
/cas/bin/client [-p num] -h host -v integer (verify)
/cas/bin/client [-p num] -h host -f file (import)
/cas/bin/client [-p num] -h host -s 'select' (export)
```

Table 5. Test program options

Usage	Description
-h host	Required. Specifies the name or dotted decimal Internet address of the Picture Perfect host running the server.
-v num	Specifies that the number, num, is to be used to verify that the Picture Perfect server is responding. The value of num also forces the debug level at the host. The debug level is set to the value (num mod 10) for the duration of the next operation. When debugging is set to a high level, detailed information about the server process is written to the Picture Perfect log. The server automatically resets the debug level to zero after the next import or export operation in order to prevent log files from growing too large.
-f input file	Specifies the name of a file containing one or more records to be imported into Picture Perfect preceded by a valid presentation and application header.
-s 'select ...'	Specifies that the string inside the quotation marks is an SQL SELECT statement. **Note:** To accommodate the user's ability to export data, modify it, then import it back into the Picture Perfect database, for the three tables which have had their categories (and in the case of the `person`, the user fields as well), stripped out, the `importd` server's export function has been modified, so an export performing a `'SELECT * FROM' person, area,` or `area_event`, will result in records from the updated table structure, plus the categories from the associated sub-table appended to the end of each record. If the user wishes to get records from just the base table, then an accommodation has been made, so, for these three tables, if the user uses a `SELECT` of the form: `'SELECT ** FROM ... '`, the additional sub-table records will not be retrieved. This feature is applicable whether the client program is being used, or the user's own RPC client program is being used.
-p num	Optional. Requests that the client connect to a specific server process at sites where multiple instances of the server are running. Each server running is invoked with a numeric parameter beginning with zero. This allows the Picture Perfect host to handle multiple requests concurrently.

Application Layer Services

Each of the three services are accessed by calling the function associated with that service. The three functions are:

```
long *client_verify_V1()
export_r *client_import_V1()
export_r *client_export_V1()
```

The results of the successful execution of these functions are interpreted based on the operation being performed. With the exception of the `verify` service, a pointer to an `export_r` data type is returned.

The structure of the `export_r` data type is shown below:

```
typedef struct node {
              char *buf;          /* Pointer to a line of data */
              struct node *next;  /* Pointer to the next item on the list */
} node;

typedef node *buflist;

typedef struct {
              int errnum;         /* Contains the host system's errno, if any */
              int cause;          /* Contains a server specific error code */
              int good;           /* Number of good transactions */
              int bad;            /* Number of error transaction */
              union {
              buflist list;       /* Linked list of returned data or error records */
              } export_r_u;
} export_r;
```

Troubleshooting

Failure to execute falls into two general categories:

- Communication failures.
- Application related errors.

Communication Failures

Communications failures cause a pointer to NULL to be returned. These errors most commonly result in the following error messages:

Table 6. Communication failure error messages

Error message	Description
`can't get internet address for 'host'`	
	The specified host is unknown to the client. Check the client's host table to ensure that it contains an entry for the Picture Perfect host.
`rpc error: the pmapper failed in its call`	
	The client host table entry for the specified host contains the wrong Internet address. Check the client host table to verify that the Internet address is correct for the Picture Perfect host. The network is not operational or the client's network interface is not up. Try to ping the Picture Perfect host using the ping command supplied with your TCP/IP software. If this is unsuccessful, then either your network interface is not configured properly or you are experiencing some other network problem. First, check your client interface to verify that it is configured for the correct type of interface (for instance, Ethernet, Token Ring) and that it is using the correct Internet address. If you are using a gateway or domain name service, the gateway and/or name servers may need to be updated with the Picture Perfect host address. Contact your network administrator for assistance with this.
`rpc error: the remote program is not registered`	

	The specified host is not running the `importd` server. To check this, log on to the Picture Perfect host as `root` and type `ps -e | grep importd`. Verify it was correctly installed by typing the following:
	`. /cas/bin/profile`
	`query tps_daemons`
	You should see entries similar to the following:
	`.`
	`.`
	`10 /cas/bin/importd 0 -tps R A 20030114 101507`
	`11 /cas/bin/importd 1 -tps R A 20030114 101507`
	`.`
	`.`
	If these entries do not exist, then there was an error during the `impexp` package installation, and it should be removed and then re-installed.
	If the entries exist, then possibly the process died. Re-start Picture Perfect, and check again to see if the any of the `importd` server processes are running. Also check the `/cas/log/log.MMDD` file to see if there is any explanation for the failure. For further help, call your GE Security technical support representative.

Application-Related Errors

Application errors will always result in the successful execution of the remote procedure call and a non-NULL `export_r` pointer, but the requested operation fails due to some application-dependent problem. An application error is indicated by a non-zero bad component. It is possible for both the good and bad components to be non-zero.

An example of an application error might be a syntax error in the SQL SELECT statement that contains an export request. Another example might be an incorrectly formatted application header. A third example might be the inclusion of a department ID in a `person` record where the department is non-existent in the Picture Perfect database.

These types of errors are slightly more difficult to diagnose, since a working knowledge of Picture Perfect's operation may be required. The best way to approach these problems is to enable full debugging using the verify function and repeat the procedure. After the operation completes, check the import log on the Picture Perfect host for further diagnostics.

Verify Service

The Verify service is used to check connectivity between the client and server application programs. It is similar to the TCP/IP `ping` command, except that it tests the communications link between the application and presentation layers, whereas the `ping` command tests the communications link at the network and link layers. Verification is performed by passing a number to the Picture Perfect server which sends the number back to the client. If the number received back from the server is the same as the number sent, it can be assumed that the server is operational.

A useful by-product of the verify call is the ability to set the server's level of debugging for the next operation. After the next operation completes, the server's debug level is automatically reset to `0`. The debug level is determined by the `modulus 10` of the least significant digit of the number sent to the server. The maximum debugging level is `9`. Debug information is written to the current day's import log file located in the `/cas/log` directory. The `import` log files are named `imp.mmdd` where `mm` is the month and `dd` is the day of month.

The types of diagnostics associated with debug levels are shown below:

Table 7. Debug level diagnostics

Debug level	Result
0	No debugging information
1	Temporary file names of imported files are logged
2-3	Input output and error files are preserved
4-5	Additional file I/O diagnostics are logged
6	Each error file line is logged (import only)
7	Each output file line is logged (import and export)
8	Each received line is logged (import and export)
9	All possible diagnostics

Note: Debug levels are cumulative--that is, a level of `7` includes the capabilities of levels `1` through `6`.

Verify Prototype

```
long *client_verify_V1 (char *host, long *argp)
```

The `host` parameter must be a character pointer to the Picture Perfect host name or its Internet address. Host names may also include the complete domain name, provided that the client's TCP/IP subsystem is configured to use a domain name server.

The `argp` parameter must be a pointer to a long integer representing the number to be sent to the server.

Verify Diagnostics

If the function completes successfully, a pointer to a long integer is returned. The integer should be inspected to ensure that it is equal to the contents at `argp`, since part of the verification should include checking byte and word ordering between machine architectures.

If the function fails, the appropriate error message is output to `stderr` and a long pointer to NULL is returned.

If the function returns a non-NULL pointer, but it points to a different value, your client software may have a problem. Contact your GE Security technical support representative for further assistance.

Import Service

The Import service is used to import data into Picture Perfect and to update the microcontrollers with pertinent changes. As explained in *Chapter 3 Importing data*, data to be imported consists of a file containing an application and presentation header followed by one or more delimited records. With RPC, however, the import file is created by the client application on the client machine.

Since RPC performs all presentation layer conversions, the data type (first byte) of the presentation header must be the character `A` and the carriage control (second byte) should be the character `0` (zero). The application header should be formatted as specified in *Header formats* on page 13.

Import Prototype

```
typedef struct {
FILE *fp;     /* Pointer to the open import file */
char *name;   /* Pointer to the name of the file */
}impexp;

export_r *client_import_V1 (char *host, impexp *args)
```

The `host` parameter must be a character pointer to the Picture Perfect host name or its Internet address. Host names may also include the complete domain name, provided that the client's TCP/IP subsystem is configured to use a domain name server.

The `args` parameter must be a pointer to an `impexp` data type whose `fp` component is a pointer to the open import file. The name component should point to a string containing the file's name.

Import Diagnostics

If the function returns successfully, a pointer to an `export_r` structure is returned. This should be inspected for application layer errors. The good component of the `export_r` structure contains the number of records successfully imported.

The bad component contains the number of records rejected for one reason or another. If the bad component is non-zero and the `errnum` component is zero, then the linked list pointed to by the `export_r_u.list` component should be traversed, which will yield one or more meaningful error messages for each rejected record. Each error message is preceded by the input file's record number and is terminated with a `newline` character. If the `errnum` component is non-zero, the server was unable to open or read either the output or error file. The `errnum` component corresponds to the server's `errno` value. The cause component indicates the file and the access mode that failed, conforming to the following conventions:

```
# define SRV_OUTFILE_OPEN_ERR     1 /* Couldn't open the output file */
# define SRV_ERRFILE_OPEN_ERR     2 /* Couldn't open the error file */
# define SRV_OUTFILE_READ_ERR     3 /* Couldn't read the output file */
# define SRV_ERRFILE_READ_ERR     4 /* Couldn't read the error file */
# define SRV_EXPORT_EXEC_ERR      5 /* Couldn't exec a database program */
```

If the RPC call fails completely, a NULL pointer is returned.

Export Service

The Export service is used to request records from the Picture Perfect database using standard `SQL SELECT` statements. The exception to this rule occurs when a select is made for all of the columns of either the `area`, `area_event`, or `person table`, for example, `SELECT * FROM person WHERE id < 100`.

Because the table structure has changed in Picture Perfect, to accommodate the user's ability to export data, modify it, then import it back into the Picture Perfect database, the `importd` server's export function has been modified, so an export performing a `'SELECT * FROM ' person, area, or area_event`, will include records from the original table structure as well as from the associated sub-tables appended to the end of each record, as shown in *Table 8*:

Table 8. Updated table structure

Original table structure	New table structure
area	area area_category
area_event	area_event area_event_cat
person	person person_user person_category badge

If the user wishes to get records from just the base `person, area,` or `area_event` table, then an accommodation has been made, so, for these three tables, if the user uses a `SELECT` of the form: `'SELECT ** FROM ... '`, the additional sub-table records will not be retrieved. This feature is applicable whether the provided client program is being used, or the user's own RPC client program is being used.

Export Prototype

```
export_r *client_export_V1 (char *host, char **statement)
```

The `host` parameter must be a character pointer to the Picture Perfect host name or its Internet address. Host names may also include the complete domain name, provided that the client's TCP/IP subsystem is configured to use a domain name server.

The `statement` parameter must be the address of a pointer to a string containing an `SQL SELECT` statement.

The length of the `SELECT` statement cannot exceed 4096 characters.

Export Diagnostics

Upon successful completion, a pointer to an `export_r` structure is returned. The good component corresponds to the number of rows selected. The bad component corresponds to the number of lines of error messages returned. It is possible for both components to be non-zero. Linked list `export_r_u.list` will always contain the selected rows followed by any error messages. Like the import service, the `errnum` and cause components should be inspected for file I/O errors on the host.

The length of a returned row will not exceed 4096 characters.

If the RPC call fails completely, a NULL pointer is returned.

Running Concurrent Clients

The Picture Perfect server process, `importd`, handles incoming requests serially. Therefore, if a client's request takes a long time to execute, another client can potentially time out waiting for the server to become available.

Server Configuration

During installation, the Picture Perfect host can be configured to start up multiple `importd` servers. Each server is started up with a unique integer argument which the server uses to register itself with the `portmapper`. The base program number for Picture Perfect RPC server is 540,000,000. If `importd` is invoked without arguments, the base program number is registered. The numeric value of the first argument is added to the base program number to yield the program number registered with the `portmapper`. For example, if the commands:

```
/cas/bin/importd 0
/cas/bin/importd 1
/cas/bin/importd 2
/cas/bin/importd 3
```

are executed on the Picture Perfect host, the program numbers 540,000,000 through 540,000,003 are registered with the **`portmapper`**.

Client Configuration

In order for the client to take advantage of this feature, it must specify which server it intends to use. This is accomplished through the manipulation of the global unsigned long integer, `impprog`, which defaults to the value of `IMPPROG`, defined in `imex.h`. To connect to the server invoked as `/cas/bin/importd 1` for example, the client program should set its impprog variable to `(IMPPROG + 1)` as in:

```
impprog = IMPPROG + 1;
```

Timeouts

The header file `imex.h` contains default time-out values for each of the three RPC services. These time-outs are used to specify the maximum allowable time in seconds an RPC may execute before the RPC is considered to fail. The import and export service can take longer than the defaults, which can cause the procedure to terminate prematurely with a time-out error. If this occurs, modify the time-out values. The defaults are currently defined in `imex.h` as follows:

```
# define VERIFY_TIMEOUT 5          /* Shouldn't take long at all */
# define IMPORT_TIMEOUT 60         /* Depends on number of records/file*/
# define EXPORT_TIMEOUT 120        /* Depends on complexity of the query*/
```

Inventory of source code supplied with Import/Export

The following files comprise the Import/Export Interface RPC Toolkit:

```
/usr/rpc/imex.h
/usr/rpc/cli_main.c
/usr/rpc/cli_verify.c
/usr/rpc/cli_import.c
/usr/rpc/cli_export.c
/usr/rpc/cli_create.c
/usr/rpc/imex_xdr.c
/usr/rpc/makefile.cli
```

Table 9. Source code file types

File type	Description
imex.h:	This is a common header used by client and server. It contains definitions of the data structures used for data transfer as well as ANSI prototypes for all functions used within the subsystem.
cli_*.c:	This is a sample program whose source code is made available to the customer. We provide a front-end to each procedure which can be customized for the particular application
imex_xdr.c	This file contains all of the custom-written XDR routines to handle data transfer. These functions are common to client and server. Although it is unlikely that any of these routines will need modification on the client machine, they have been provided to aid in porting to new architectures.

Chapter 6 Troubleshooting and support

This section provides information to help you diagnose and solve various problems that may arise while configuring or using your GE product and offers technical support contacts in case you need assistance.

In this chapter:

Error Messages 48
Contacting technical support 53

Error Messages

Error messages that you may encounter while using the Import/Export program are explained below and listed in alphabetical order:

Table 10. Communication failure error messages

Error message	Description
Bad date, must be YYYYMMDD	
	The date must be in the format of year month day.
Bad time, must be 0..235959	
	The time must be within the range of 000000 to 235959.
Badge for employee `employee num` has no BID	
	The record on the host badge table does not have a BID. Check the employee number to see if it is correct.
Blank field, requires data	
	This particular field requires data and was left blank.
Can't find system_config record for setup	
	Check the environment and make sure Picture Perfect is running.
Can't get internet address for `host`	
	The specified host is unknown to the client. Check the client's host table to ensure that it contains an entry for the Picture Perfect host.
Can't open filename, errno `num`	
	Error occurred when trying to open the filename to read.
Can't read presentation header, errno `num`	
	Error occurred trying to read the presentation header.
Can't update department `description` using id `num`, not in database	
	Updating a department requires an id in the `department.id` column.
Can't update department with no database ID	
	Updating a department requires an id in the `department.id` column.
Can't update micro - TPS is not running	
	Picture Perfect must be running in order to update the micros.
Can't update operator activity. TPS not running	
	Picture Perfect must be running in order to update operator activity history.
Data is too long	
	The value of the column is too long for the field.

Table 10. Communication failure error messages

Error message	Description
Department description column is required for update	
	You must use a description in the proper field.
Employee `employee num` has multiple badges	
	The badge record selected for updating based on `employee num` has more than one record; therefore, the record cannot be updated. You must update the desired record through Picture Perfect.
Error accessing process control in TPS	
	Make sure Picture Perfect is up and running, then try again.
Error fetching hostname for network insert for host id	
	The host table is not setup correctly and you are in a networked system. Verify by using the Control/Host options from the Picture Perfect menu.
Error importing data	
	There is an error in the data you are trying to import. Verify the data you are importing.
Error, incorrect header layout	
	The header in the import file contains an error. Verify each byte to make sure it contains the correct layout. login_id= *login name* data type = *num* table = *table name*
Error opening error file	
	Check the name and permissions of the error file designated in the header or on the command line.
Error opening error file `filename` for appending	
	Check the name and permissions of the error file designated in the header or on the command line.
Error opening error file `filename` for writing	
	Check the name and permissions of the error file designated in the header or on the command line.
Error opening import file	
	The filename specified does not exist or cannot be opened.
Error opening output file	
	The filename specified does not exist or can not be opened. Check the name and permissions of the filename.
Error opening output file `filename` for appending	
	Check the permissions of the filename.
Error opening output file `filename` for writing	
	Check the permissions of the filename.

Table 10. Communication failure error messages

Error message	Description
Error, you may not insert or update a category, department or permission_group table from a subhost in a network	
	The network host is the only place you may insert or update one of the above listed tables if your system is configured as an enterprise version of Picture Perfect.
Field required for import	
	The column identified requires data and it does not contain any.
Field value is too big	
	The value exceeds the limit of the field.
Header error - error file name is required	
	The header record must contain an error file name.
Header error - invalid signature	
	The format of your header does not match the required layout.
Header error - output file name is required	
	The header record must contain an output file name.
Header error - unknown transaction type	
	The transaction type must be a valid one listed in the manual, such as 31 or 32.
Header option requires id for department `description`	
	The header is indicating that the import wants to use a department description in the `person.department` column and it is not there.
Id column is required for update	
	The id of the record is needed to do an update and the import program was unable to retrieve it. Put it in the import file and try again.
ID is null for department `dept name`	
	Id is required and there is none in the **`id`** field of the import file.
Illegal duplicate insert attempted	
	The record has already been inserted into Picture Perfect.
Import file `filename` rejected, no transactions processed	
	There is an error locating or using the filename specified with `importdb.` Verify that the filename is correct and that it exists.
Import file `filename` rejected, unknown transaction type `type`	
	The transaction type must be a valid one listed in the manual, such as 31 or 32.
Import rejected on Backup host. Perform import on Primary host	
	Import operations are permitted only on the primary host in a redundant Picture Perfect system.
Invalid data in field	

Table 10. Communication failure error messages

Error message	Description
	The data for the field contains an error. Check your data.
Invalid data is causing a syntax error	
	Refer to the statement in the input file and check your data.
Invalid data is causing SQL error `num`	
	Use the utility program `sqlmsgnbr num` to obtain more information regarding the SQL error.
Invalid presentation header size - `num`	
	The format of your header does not match the required layout.
Invalid table `table name` in header is causing SQL error	
	Check the spelling of the table name.
Invalid table `table name` in header record	
	Check the spelling of the table name.
Memory allocation error	
	The import records are most likely coming in too fast for Picture Perfect. Re-start Picture Perfect and send the records in slower. This can happen on updates, since the badge update must also update all the micros that contain the badge.
No operator login_id for employee `employee num`	
	Verify that the username or operator `login_id` specified in the header actually exists as an operator in Picture Perfect.
No operator permission	
	The import is trying to update or insert a record containing a category that the operator does not have permission to use. From the menu, check the Picture Perfect permissions and permission group setup for the operator. The operator for the import is specified in the header record.
No operator record found with employee_id `employee num`	
	Verify that an operator exists in Picture Perfect based on the `employee num.`
No permission for operator user name to assign category description in field category position	
	You are trying to insert a category on a badge and you do not have operator permission set up in Picture Perfect for that `category` or `id`, based on the operator `user_name` in the import file.
No transactions processed	
	An error was encountered and no transactions were inserted or updated.
Record #: invalid badge format in field 'bid', data = bid #	
	The `bid` number does not match the badge format specified in `system_config.badge_id_format.`
Record #: value is too long in field 'bid', data = bid #	
	The **`bid`** number contains more numbers than allowed according to `system_config.badge_id_format.`

Table 10. Communication failure error messages

Error message	Description
Record #: value is too short in field 'bid', data = bid #	
	The `bid` number contains fewer numbers than allowed according to `system_config.badge_id_format`.
Related table requires data	
	The particular column identified is a foreign key to another table. Data must be inserted into the other table before it can be referenced.
Rpc error: the pmapper failed in its call	
	The client host table entry for the specified host contains the wrong Internet address. Check the client host table to verify that the Internet address is correct for the Picture Perfect host.
Rpc error: the remote program is not registered	
	The specified host is not running the `importd` server.
Shared memory allocation error	
	You are sending in updates with operator history logging too fast for the system to process. Restart Picture Perfect and send your import transactions in slower.
SQL error `num` connecting to database	
	Verify that Picture Perfect is up and running, then check your environment to make sure it can reach the database.
Too few fields in record	
	Check the import record to see if it has less fields than the table in Picture Perfect. Verify that the table layout is followed.
Too many fields in record	
	Check the import record to see if it has more fields than the table in Picture Perfect. Verify that the table layout is followed.
Total valid transactions = *num* Total error transactions = *num* Total transactions = *valid* + *error*	
	This is the summary report of what happened at the end of the import.
TPS must be running - operation not allowed	
	Verify that Picture Perfect is running.
Update attempted on non-existent record	
	Check the `id` of the record to make sure it exists in Picture Perfect.
Update not allowed on badge with invalid status 4, 5, 6	
	The status update byte in the import header is set to allow only badges with a status of "valid" to be updated. If you want to update badges with a status of 4, 5, or 6, change the header.

Contacting technical support

For assistance installing, operating, maintaining, and troubleshooting this product, refer to this document and any other documentation provided. If you still have questions, you may contact technical support and/or pre-sales.

For pre-sales and technical support assistance, we provide customers with several options (see *Table 11*). Our support phone number is available Monday through Friday, 8 a.m. to 7 p.m. Eastern Time. Protection plans are available for extended coverage.

Table 11. Pre-sales and support contact information

	Pre-sales	Technical support
Phone:	1 800 428 2733	1 888 GE SECURITY (437 3287)
Fax:	561 998 6160	561 998 6224
E-mail:	None	rs-bctsupport@ge.com

Note: Be ready at the equipment before calling for technical support.

Online publication library

Another great resource for assistance with your GE product is our online publication library, available to all of our customers. To access the library, go to our website at the following location:

`http://www.gesecurity.com`

In the **Tools** area at the top, click the *Publication Library* link. After you register and log on, you may search through our online library for the documentation you need.

Appendix A Table layouts

This appendix provides layouts for the various Picture Perfect tables.

In this appendix:

Notes 56
Alarm History (export only) 57
Person 59
Badge_History (export only) 62
Badge 66
Category 69
Facility 69
Department 70
Host BID Format 71
Operator History (export only) 72
Person_user 73
Person_category 74
Area 75
Area_event 77
Area_category 80
Area_event_category 81

Notes

Note: In some cases, certain fields require data in order to import the record successfully. In the following tables, required fields are marked with a double asterisk (**). Fields which may be required, depending on transaction type, are marked with a single asterisk (*).

Note: When populating the `badge` table from a one-time download, it will be absolutely necessary to populate the `host_bid_format` tables before attempting to insert any badge records. In addition, if the badge record contains a value in the `bid_format_id` or `facility` field, then the Picture Perfect `host_bid_format` and `facility` tables must be populated first, as well.

Note: When populating the `person` table, the format of the record being inserted MUST include the pipe `(||)` delimiters at the end of the basic structure to accommodate the possibility of 40 user fields, 96 categories, and the 31 fields of a badge record, even if there is no accompanying data for the person record. The file, `sample.imp`, in the `/cas/import` directory specifies the format required to import a complete Person record with a category and badge. An example follows:

Figure 6. Table Layout Example 1

```
||0|1||||4334||Jake|Smith|JS||22|||||||0|||||||||||||-1|20041213|220905|User field 1||||||||||
|||||||||||||||||||||||||||||User field 40|-1|||||||||||||||||||||||||||||||||||||||||||||||||||
|||||||||||||||||||||||||||||||||||||||||-1|||||||||||||||||||||||||||||||||||
```

The above example shows a person record with the first (User field 1) and fortieth (User field 40) user field populated and the first (-1) and ninety-sixth (-1) category field populated and no data in the badge portion of the record (the last 31 fields.) The total number of fields for a person record is 203.

If an update transaction type 31 (Update existing records in any table with an operator-activity transaction for each modification) is being performed, then the required extra `employee number` field will occur after the person record, and before the required additional 40 user fields, 96 categories, and the 31 fields of a badge record. An example follows:

Figure 7. Table Layout Example 2

```
4||0|1||||4634||James|Smith|JS||23|||||||0|||||||||||||-1|20041213|220905|0004|User field 1||||
|||||||||||||||||||||||||||||||||||User field 40|-1|||||||||||||||||||||||||||||||||||||||||||
|||||||||||||||||||||||||||||||||||||||||||||||||||-1|||||||||||||||||||||||||||||||||||
```

The above example shows the employee number (0004) inserted at the end of the person record data and just before the first person_user field (User field 1.)

Note: When populating the `area,` or `area_event` tables, the format of the record being inserted MUST include the pipe `(||)` delimiters at the end of the basic structure to accommodate the possibility of an extra 32 categories, even if no categories are being assigned. If an update transaction type 31 (Update existing records in any table with an operator-activity transaction for each modification) is being performed, then the required extra `employee number` field will occur after the `area` or `area_event` record, and before the required additional 32 `category` fields.

Alarm History (export only)

Table 12. Alarm History (export only)

COL #	Column name	Range value	Length	Meaning
1	id	1 to +2,147,483,647	11	Unique Id Unique record number assigned by system
2	input_group	1 to +2,147,483,647	11	Input Group Foreign key to id field in input_group table
3	input_desc	1 to 60 bytes	60	Input description
4	alarm_desc	1 to 60 bytes	60	Alarm description
5	micro	1 to 4096	20	Micro Micro address
6	operator	1 to 8 bytes	8	Operator Responding operator login name
7	bid	1 to 16 bytes	16	Badge Encode Number Responding operator badge encode number
8	physical_state	1 to 5	8	Input State 1=Short; 2=Cut; 3=Open; 4=Closed; 5=N/A
9	logical_state	1 to 3	7	Condition 1=Alarm; 2=Reset; 3=Tamper
10	process_state	1 to 8	9	Process State 1=Active; 2=Bumped; 3=Notified; 4=Remote Notified; 5=Pending; 6=Completed; 7=Removed; 8=Purged
11	priority	1 to 500	6	Priority Alarm priority
12	response	1 to 255 bytes	255	Response Alarm responses selected
13	opr_respond_date	>= 19000101	11	Respond Date Form yyyymmdd in the time zone of the operator
14	opr_respond_time	0 to 235959	11	Respond Time Form hhmmss in the time zone of the operator
15	host_respond_date	>= 19000101	11	Host Respond Date Form yyyymmdd in the time zone of the host
16	host_respond_time	0 to 235959	11	Host Respond Time Form hhmmss in the time zone of the host
17	utc_respond_date	>= 19000101	11	UTC Respond Date Form yyyymmdd in UTC/GMT time
18	utc_respond_time	0 to 235959	11	UTC Respond Time Form hhmmss in UTC/GMT time
19	happened_date	>= 19000101	11	Happened Date Form yyyymmdd in the time zone of the operator

Table 12. Alarm History (export only)

COL #	Column name	Range value	Length	Meaning
20	happened_time	0 to 235959	11	Happened Time Form hhmmss in the time zone of the operator
21	host_happened_date	>= 19000101	11	Host Happened Date Form yyyymmdd in the time zone of the host
22	host_happened_time	0 to 235959	11	Host Happened Time Form hhmmss in the time zone of the host
23	utc_happened_date	>= 19000101	11	UTC Happened Date Form yyyymmdd in UTC/GMT time
24	utc_happened_time	0 to 235959	11	UTC Happened Time Form hhmmss in UTC/GMT time
25	facility	-1 to +2, 147, 483, 647	11	Facility Foreign key refers to id field in facility table Record number of the facility this record is in

Person

Table 13. Person

COL#	Column name	Range value	Length	Meaning
1	id	1 to +2,147,483,647	11	id Unique record number assigned by system
2	pin	1 to 10 bytes	10	Pin Keypad personal ID number
3	status**	0,4,5,6	6	Status Badge status: 0=Active; 4=Lost; 5=Suspended; 6=Deleted
4	type**	1 to the number of records in the person_type table	11	Type Foreign key to id field in person_type table
5	person_kp_resp	1to the number of records in the person table	11	Keypad Response Foreign key to id field in person table
6	person_trace	0, 1	11	Card Trace Enable Person Trace 0=No; 1=Yes
7	person_trace_alarm	0, 1	12	Generate Alarm when Person Traced If Person Trace is enabled, an alarm is generated every time the badge is read. 0=No; 1=Yes
8	employee	1 to12 bytes	12	Employee ID Employee number
9	department	1 to the number of records in the department table	11	Department Foreign Key id to department table with header byte 123 = I; 30 bytes for department description with header byte = D
10	first_name**	1 to 40 bytes	40	First Name Badgeholder's first name
11	last_name**	1 to 40 bytes	40	Last Name Badgeholder's last name
12	initials**	1 to 6 bytes	6	Initials Badgeholder's initials
13	title	1 to 20 bytes	20	Title Badgeholder's title
14	address1	1 to 40 bytes	40	Address 1
15	address2	1 to 40 bytes	40	Address 2
16	address3	1 to 40 bytes	40	Address 3
17	address4	1 to 40 bytes	40	Address 4
18	address5	1 to 40 bytes	40	Address 5

Table 13. Person

COL#	Column name	Range value	Length	Meaning
19	phone	1 to 30 bytes	30	Phone 1 Primary phone number
20	phone2	1 to 30 bytes	30	Phone 2 Alternate phone number
21	reissue_cnt	1 to 99	2	Reissue Count The number of times the badge has been issued
22	apb	0,1,2,3	11	Anti-passback Should be set to 0 for new badge inserts. Antipassback status: 0=Neutral; 1=In; 2=Out; 3=Privileged
23	reader	1 to +2,147,483,647	11	Reader The reader that last granted access to this person
24	access_date	>= 19000101	11	Access Date Date in the form yyyymmdd. Automatically supplied by the system
25	access_time	0 to 235959	11	Access Time Date in the form yyyymmdd. Automatically supplied by the system
26	access_tz	0	11	Time Zone UTC. Automatically supplied by the system
27	active_date	>= 19000101	11	Activation Date Date in the form yyyymmdd. Automatically supplied by the system if left blank during inserts.
28	active_time	0 to 235959	11	Activation Time Time in the form hhmmss. Automatically supplied if left blank during inserts.
29	active_context	0, 1, 2	11	Time Context Time zone context: 0=Host, 1=Micro, 2=Operator
30	deactive_date	>= 19000101	11	Deactivation Date Date in the form yyyymmdd.
31	deactive_time	0 to 235959	11	Deactivation Time Time in the form hhmmss.
32	deactive_context	0, 1, 2	11	Time Context Time zone context: 0=Host, 1=Micro, 2=Operator
33	force_download	0	11	Download Upon Save Set to 0. Not used by Import/Export.
34	facility**	-1 to +2,147,483,647	11	Facility Foreign key refers to id field in facility table. Record number of the facility this record is in.

Table 13. Person

COL#	Column name	Range value	Length	Meaning
35	modify_date	>= 19000101	11	Modify Date Date in the form yyyymmdd. System will supply the current date if no date is provided.
36	modify_time	0 to 235959	11	Modify Time Time in the form hhmmss. System will supply the current time if no time is provided.
37 - 76	user1 - user40	40 bytes each	40	User comment column. There are 40 user fields, user1 through user40.
77 - 172	Category1 - Category96	1 to +2,147,483,647	11	Foreign keys to id's in category table. There are 96 category fields, category 1 through category 96.
173-203	Badge fields[1]	*1*	*1*	The badge portion of the record.1

1. For information on the badge fields, see *Badge* on page 66.

Badge_History (export only)

Table 14. Badge History (export only)

COL #	Column name	Range value	Length	Meaning
1	id	1 to +2,147,483,647	11	Unique Id Unique record number assigned by system.
2	bid	1 to 16 bytes	16	Badge Encode Number
3	pin	1 to 10 bytes	10	Pin Keypad personal ID number.
4	employee	1 to 12 bytes	12	Employee ID Employee number
5	first_name	1 to 40 bytes	40	First Name Badgeholder's first name.
6	last_name	1 to 40 bytes	40	Last Name Badgeholder's last name.
7	area	1 to the maximum record number in the area table	11	Area Foreign key to id field in area table. Record number that the reader is in
8	category	-1 to the maximum record number in the category table	11	Category Foreign key to id field in category table. Category record number that granted access.
9	xact_type	0 to 45	17	Transaction type:
				0=Open
				1=Reader Unknown - Micro
				2=Reader Offline
				3=Reader Type Unknown
				4=Area Unknown to Micro
				5=Area Offline
				6=No Category Match
				7=Invalid APB In
				8=Invalid APB Out
				9=Invalid PIN Entered
				10=Unknown Badge
				11=Lost Badge
				12=Suspended Badge
				13=Deleted Badge
				14=Learn Timeout

Table 14. Badge History (export only)

COL #	Column name	Range value	Length	Meaning
				15=Invalid T/A In Attempt
				16=Invalid T/A Out
				17=Invalid Shunt
				18=Expired Badge
				19=APB In
				20=APB Out
				21=T/A In
				22=T/A Out
				23=Passive APB In
				24=Passive APB Out
				25=Invalid Func Code
				26=KP Resp Invalid Badge
				27=KP Invalid Open Door
				28=KP Resp Not Enabled
				29=Not Validated
				30=Degraded Open
				31=Usage Exhausted
				32=Valid No Passage
				33=2nd Badge Door Lock
				34=Valid Toggle Reader
				35=Valid - Door Was Not Locked
				36=Tour Badge
				37=Valid - No Door
				38=Swipe and Show
				39=Valid Floor
				40=Invalid Floor
				41=N/A
				42=Valid Timed APB In
				43=Invalid Timed APB In
				44=Valid Timed APB Out
				45=Invalid Timed APB Out

Table 14. Badge History (export only)

COL #	Column name	Range value	Length	Meaning
10	duress	Y or N	1	Duress
11	shunt	Y or N	1	Shunt
12	reader_type_log	0, 8, 16, 32, 64, 128, 256, 512, 1024, 2048	6	Logical Reader Type 0=Normal; 8=Elevator; 16=APB In; 32=APB Out; 64=Console; 128=Double Badge; 256=T/A In; 512=T/A Out; 1024=Authorization Required; 2048=No Authorization Required
13	reader_type_phy	1, 2, 4	17	Physical Reader Type 1=Badge Only; 2=Badge & Keypad; 4=Keypad Only
14	reader_desc	1 to 60 bytes	60	Reader Reader description.
15	dept	1 to the number of records in the department table	11	Department Badgeholder's department record number.
16	status	0, 4, 5, 6	11	Status Badge status. 0=Active; 4=Lost; 5=Suspended; 6=Deleted
17	dev_xact_date	>= 19000101	11	Transaction Date Form yyyymmdd in the time zone of device
18	dev_xact_time	0 to 235959	11	Transaction Time Form hhmmss in the time zone of device
19	host_xact_date	>= 19000101	11	Host Transaction Date Form yyyymmdd in the time zone of host
20	host_xact_time	0 to 235959	11	Host Transaction Time Form hhmmss in the time zone of the host
21	utc_xact_date	>= 19000101	11	UTC Date Form yyyymmdd in UTC/GMT time
22	utc_xact_time	0 to 235959	11	UTC Time Form hhmmss in UTC/GMT time
23	nbadge	1 or 2	16	Number of Badges Number of badges in transaction. 1=Single; 2=Double
24	sequence	1 or 2	6	Badge Sequence 1=1st Badge in Transaction; 2=2nd Badge in Transaction.
25	kp_alarm_resp	0, 1	21	Keypad Alarm Response 0=Normal; 1=Keypad
26	personnel_type	1 to number of records in personnel_type table	11	Personnel Type Foreign key to id field in personnel_type table
27	swipe_show	0 or 1	21	Swipe and Show 0=Normal, 1=Swipe and Show

Table 14. Badge History (export only)

COL #	Column name	Range value	Length	Meaning
28	floor_accessed	0 to 64	14	Floor Accessed Floor accessed on an elevator
29	tzcorr	-24 to +24	6	Time Zone Correction Time zone offset of micro relative to the host location
30	facility	-1 to +2, 147, 483, 647	11	Facility Foreign key refers to id field in facility table. Record number of the facility this record is in.

Badge

Table 15. Badge

COL #	Column name	Range value	Length	Meaning
1	id	1 to +2,147,483,647	11	Id Unique record number assigned by system.
2	description	1 to 60 bytes	60	Badge Description.
3	bid**	1 to 16 bytes	16	Badge Encode Number
4	status	0,4,5,6	6	Status Badge status: 0=Active; 4=Lost; 5=Suspended; 6=Deleted
5	badge_tour	0, 1	11	Badge Tour Designates a badge as a tour badge used to conduct guard tours of a facility at specified intervals. This badge will not operate for normal access control 0=Disabled; 1=Enabled
6	badge_temp	0, 1	11	Temporary Badge Designates a badge as a temporary badge. A pool of badges can be created and used repeatedly for this purpose 0=No; 1=Yes
7	person_id	1 to 11 bytes	11	Person Id **Note:** If Import/Export is used to unassign a badge from a person, this field must be set to 0 (zero). If it is set to NULL, the operation will fail.
8	reader	1 to +2,147,483,647	11	Reader The reader that last granted access to this badge
9	access_date	>= 19000101	11	Access Date Date in the form yyyymmdd. Automatically supplied by the system.
10	access_time	0 to 235959	11	Access Time Date in the form yyyymmdd. Automatically supplied by the system.
11	access_tz	1 to the number of records in the time_zone table	11	Access Time Zone Time zone of last access. Automatically supplied by the system.
12	issue_date	>= 19000101	11	Issue Date Date in the form yyyymmdd. Automatically supplied by the system if left blank during inserts.
13	issue_time	0 to 235959	11	Issue Time Time in the form hhmmss. Automatically supplied if left blank during inserts.
14	issue_context	0, 1, 2	11	Issue Context Time zone context: 0=Host, 1=Micro, 2=Operator

Table 15. Badge

COL #	Column name	Range value	Length	Meaning
15	expired_date	>= 19000101	11	Expiration Date Date in the form yyyymmdd.
16	expired_time	0 to 235959	11	Expiration Time Time in the form hhmmss.
17	expired_context	0, 1, 2	11	Expired Context Time zone context: 0=Host, 1=Micro, 2=Operator
18	return_date	>= 19000101	11	Return Date Date in the form yyyymmdd
19	return_time	0 to 235959	11	Return Time Time in the form hhmmss
20	return_tz	1 to the number of records in the time_zone table	11	Return Time Zone Time zone of return
21	usage_count	1 to +2,147,483,647	11	Usage Count Number of badge swipes allowed before badge becomes exhausted. Swipes through limited usage reader types are the only ones that decrement this field.
22	usage_exhausted	0, 1	6	Usage Exhausted 0=N/A; 1=Exhausted
23	tour_badge	0, 1	11	Tour Assigned Internal Use Only for Tour Badge: 0=No; 1=Yes (if tour package is installed)
24	bid_format_id	1 to the number of entries in the host_bid_format table	11	Badge Encode Format Foreign key refers to id field in **`host_bid_format`** table. If left blank on Insert, it will automatically be assigned to the default badge id format set in the **`system_config`** table using the **Setup/System Parameters** form.
25	reissue_cnt	1 to 99	2	Reissue Count Used for seed counter; Number of times a badge is re-issued. DO NOT PUT DATA IN THIS FIELD.
26	reprint_cnt	1 to 99	2	Reprint Count Used for seed counter; Number of times a badge is printed. DO NOT PUT DATA IN THIS FIELD.
27	unique_id	1 to 16 bytes	16	Unique Id Used for seed counter; Unique id assigned to each badge, comprised of the WorkStation id, and sequential counter. DO NOT PUT DATA IN THIS FIELD.
28	badge_design	1 to the number of entries in the badge_design table	11	Badge Design Designates a badge design to be used when printing this badge

Table 15. Badge

COL #	Column name	Range value	Length	Meaning
30	facility**	-1 to +2,147,483,647	11	Facility Foreign key refers to **`id`** field in **`facility`** table. Record number of the facility this record is in
31	modify_date	>= 19000101		Modify Date Date in the form yyyymmdd. System will supply the current date if no date is provided
32	modify_time	0 to 235959		Modify Time Time in the form hhmmss. System will supply the current time if no time is provided

Category

Table 16. Category

Column name	Range value	Length	Meaning
id*	1 to +2,147,483,647	11	Id Unique record number assigned by system except when using transaction type 35
description**	1 to 60 bytes	60	Description
permission_grp	1 to +2,147,483,647	6	Permission Group Foreign key to id field in permission_group table; use value previously set up in Picture Perfect by System Administrator
m2mr_type	0, 1, 2, 3	6	M2MR Personnel Type Modified Two Man Rule Personnel Type. 0=None, 1=Team Member, 2=Guest, 3=Escort Required
facility	-1 to +2,147,483,647	11	Facility Foreign key to id field in facility table. Record number of the facility this record is in
modify_date	>= 19000101	11	Modify Date Modification date in the form yyyymmdd. System will supply the current date if no date is provided
modify_time	0 to 235959	11	Modify Time Modification time in the form hhmmss. System will supply the current time if no time is provided

Facility

Table 17. Facility

Column name	Range value	Length	Meaning
id*	1 to +2,147,483,647	11	Id Unique record number assigned by system except when using transaction type 35
description**	1 to 60 bytes	60	Description
modify_date**	>= 19000101	11	Modify Date Modification date in the form yyyymmdd. System will supply the current date if no date is provided
modify_time**	0 to 235959	11	Modify Time Modification time in the form hhmmss. System will supply the current time if no time is provided

Department

Table 18. Department

Column name	Range value	Length	Meaning
id*	1 to +2,147,483,647	11	Record number. Assigned by system except when using transaction type 35
description**	1 to 60 bytes	60	Description
division	1 to 3 bytes	3	Division Customer's division number
location	1 to 20 bytes	20	Location Geographic location
manager	1 to 24 bytes	24	Manager Manager's name
phone	1 to 30 bytes	30	Phone Telephone number
user1	1 to 60 bytes	60	User 1
user2	1 to 60 bytes	60	User 2
facility	-1 to +2,147,483,647	11	Facility Foreign key to id field in facility table. Record number of the facility this record is in
modify_date	>= 19000101	11	Modify Date Modification date in the form yyyymmdd. System will supply the current date if no date is provided
modify_time	0 to 235959	11	Modify Time Modification time in the form hhmmss. System will supply the current time if no time is provided

Host BID Format

Table 19. Host BID Format

Column name	Range value	Length	Meaning
id*	1 to +2,147,483,647	11	Id Unique record number assigned by system except when using transaction type 35
description**	1 to 60 bytes	60	Description
bid_format	1 to 22 bytes	22	Badge Id Format C-style format string used to verify and display BID
facility	-1 to +2,147,483,647	11	Facility Use -1 when inserting a new badge record
modify_date**	>= 19000101	11	Modify Date Modification date in the form yyyymmdd. System will supply the current date if no date is provided
modify_time**	0 to 235959	11	Modify Time Modification time in the form hhmmss. System will supply the current time if no time is provided

Operator History (export only)

Table 20. Operator History (export only)

Column name	Range value	Length	Meaning
id	1 to +2,147,483,647	11	Id Unique record number assigned by system
employee	1 to 12 bytes	12	Employee ID Employee ID number
login_id	1 to 8 bytes	8	Login Name Operator UNIX login name
terminal	1 to +2,147,483,647	11	Terminal Terminal ID number
xact_type	1 to 9	15	Transaction type 1=Login; 2=Logout; 3=Update; 4=Delete; 5=Query; 6=Command Event; 7=Status Request; 8=Shutdown; 9=Insert.
table_id	1 to 100	11	Table Table number affected
record_desc	1 to 60 bytes	60	Record Description Description column of affected record
field_id	1 to +2,147,483,647	11	Field Column number affected (starting with column 0)
value	1 to 60 bytes	60	Value New value
opr_activity_date	>= 19000101	11	Date Transaction date in the form yyyymmdd. (Time zone context of operator)
opr_activity_time	0 to 235959	11	Time Transaction time in the form hhmmss. (Time zone context of operator)
host_activity_date	>= 19000101	11	Host Activity Date Form yyyymmdd in the time zone context of host
host_activity_time	0 to 235959	11	Host Activity Time Form hhmmss in the time zone context of host
utc_activity_date	>= 19000101	11	UTC Activity Date Form yyyymmdd in UTC/GMT time
utc_activity_time	0 to 235959	11	UTC Activity Time Form hhmmss in UTC/GMT time
facility	-1 to +2,147,483,647	11	Facility Foreign key to id field in facility table. Record number of the facility this record is in

Person_user

Table 21. Person_user

COL #	Column name	Range value	Length	Meaning
1	id*	1 to +2,147,483,647	11	Id Unique record number assigned by system except when using transaction type 35
2	description	1 to 60 bytes	60	Description User field description/value
3	person_id	1 to maximum record number in person table	11	Person Id Foreign key to id field in person table. Record number of the person the user field is in
4	slot_number	1 to 40	11	Slot Number Slot number the user field is assigned to (the user field number)
5	facility	1 to +2,147,483,647	11	Facility Foreign key to id field in facility table. Record number of the facility this record is in
6	modify_date	>= 19700101	11	Modify Date Modification date in the form yyyymmdd. System will supply the current date if no date is provided
7	modify_time	0 to 235959	11	Modify Time Modification time in the form hhmmss. System will supply the current time if no time is provided

Person_category

Table 22. Person_category

COL #	Column name	Range value	Length	Meaning
1	id*	1 to +2,147,483,647	11	Id Unique record number assigned by system except when using transaction type 35
2	person_id	1 to maximum record number in person table	11	Person Id Foreign key to id field in person table. Record number of the person the category is in
3	category_id	1 to maximum record number in category table	11	Category Id Foreign key to id field in category table. Record number of the category on the badge with badge_id
4	slot number	1 to 96	11	Slot Number Slot number the user field is assigned to (the user field number)
5	facility	1 to +2,147,483,647	11	Facility Foreign key to id field in facility table. Record number of the facility this record is in
6	modify_date	>= 19700101	11	Modify Date Modification date in the form yyyymmdd. System will supply the current date if no date is provided
7	modify_time	0 to 235959	11	Modify Time Modification time in the form hhmmss. System will supply the current time if no time is provided

Area

Table 23. Area

COL #	Column name	Range value	Length	Meaning
1	id*	1 to +2,147,483,647	11	Id Unique record number assigned by system except when using transaction type 35
2	description	1 to 60 bytes	60	Description User field description/value
3	valid_routing	1 to +2,147,483,647	11	Valid Routing Foreign key to id field in the routing table. Record number of the type of routing to do for valid badge reads for an area
4	inv_routing	1 to +2,147,483,647	11	Invalid Routing Foreign key to id field in the routing table. Record number of the type of routing to do for invalid badge reads for an area
5	susp_routing	1 to +2,147,483,647	11	Suspended Routing Foreign key to id field in the routing table. Record number of the type of routing to do for suspended badge reads for an area
6	lost_routing	1 to +2,147,483,647	11	Lost Routing Foreign key to id field in the routing table. Record number of the type of routing to do for lost badge reads for an area
7	unk_routing	1 to +2,147,483,647	11	Unknown Routing Foreign key to id field in the routing table. Record number of the type of routing to do for unknown badge reads for an area
8	apb_routing	1 to +2,147,483,647	11	Antipassback Routing Foreign key to id field in the routing table. Record number of the type of routing to do for anti-passback badge reads for an area
9	escort_routing	1 to +2,147,483,647	11	Escort Routing Foreign key to id field in the routing table. Record number of the type of routing to do for escort badge reads for an area
10	inv_input_group	1 to +2,147,483,647	11	Invalid Grp Foreign key to id field in input_group table. Record number of the input group for an invalid badge alarm read for an area
11	susp_input_group	1 to +2,147,483,647	11	Suspended Grp Foreign key to id field in input_group table. Record number of the input group for a suspended badge alarm read for an area
12	lost_input_group	1 to +2,147,483,647	11	Lost Grp Foreign key to id field in input_group table. Record number of the input group for a lost badge alarm read for an area
13	unk_input_group	1 to +2,147,483,647	11	Unknown Grp Foreign key to id field in input_group table. Record number of the input group for an unknown badge alarm read for an area
14	apb_input_group	1 to +2,147,483,647	11	Antipassback Grp Foreign key to id field in input_group table. Record number of the input group for an anti-passback badge alarm read for an area

Table 23. Area

COL #	Column name	Range value	Length	Meaning
15	duress_input_group	1 to +2,147,483,647	11	Duress Grp Foreign key to id field in input_group table. Record number of the input group for a duress badge alarm read for an area
16	escort_input_grp	1 to +2,147,483,647	11	Escort Input Group Foreign key to id field in input_group table. Record number of the escort input group this area is a member of
17	permission_group	1 to +2,147,483,647	6	Foreign key to id field in permission_group table. Record number of the permission group this area is a member of
18	logical_state	0, 1	6	Logical State 0=Offline, 1=Online
19	physical_state	0, 2	6	Physical State 0=Disabled, 2=Enabled
20	shunting	0, 4	6	Shunting 0=Disabled, 4=Enabled
21	scheduling	0, 8	6	Scheduling 0=Disabled, 8=Enabled
22	apb_enforcement	0, 16	6	APB Enforcement 0=Normal, 16=Passive
23	route_def	1 to +2,147,483,647	11	Route Definition Foreign key to id field in route_def table. Record number of the route definition for Activity Monitor routing.
24	m2mr_door_rel_to	0 to 32,767	6	M2MR Door Release Timeout
25	occupancy_count	1 to +2,147,483,647	11	Occupancy Count Number of occupants in the area.
26	two_man_rule	0 to 4	6	Two Man Rule 0=Do not care, 1=Disabled, 2=Standard, 3=Modified Door Control, 4=Modified No Door Control.
27	occupancy_ctrl	0 to 2	6	Occupancy Control 0=Do not care, 1=Disabled, 2=Enabled
28	facility	1 to +2,147,483,647	11	Facility Foreign key to id field in facility table. Record number of the facility this record is in.
29	modify_date	>= 19700101	11	Modify Date Modification date in the form yyyymmdd. System will supply the current date if no date is provided.
30	modify_time	0 to 235959	11	Modify Time Modification time in the form hhmmss. System will supply the current time if no time is provided.
31 - 62	category1 - category32	1 to +2,147,483,647	11	Foreign keys to id's in category table. There are up to 32 category fields on an area record.

Area_event

Table 24. Area_event

COL #	Column name	Range value	Length	Meaning
1	id*	1 to +2,147,483,647	11	Id Unique record number assigned by system except when using transaction type 35
2	description	1 to 60 bytes	60	Description Area event description
3	mode_id	1 to +2,147,483,647	11	Mode Foreign key to id field in modes table. Record number of the mode this event is a part of
4	time_of_day	0 to 235959	11	Time of Day Event time in the form hhmmss. Time of day the event will run
5	context	0, 1, 2	11	Context Time zone context 0=Host, 1=Micro, 2=Operator
6	sunday	0, 1	6	Sun Indicates if the event is to run on this day 0=No, 1=Yes
7	monday	0, 1	6	Mon Indicates if the event is to run on this day 0=No, 1=Yes
8	tuesday	0, 1	6	Tue Indicates if the event is to run on this day 0=No, 1=Yes
9	wednesday	0, 1	6	Wed Indicates if the event is to run on this day 0=No, 1=Yes
10	thursday	0, 1	6	Thu Indicates if the event is to run on this day 0=No, 1=Yes
11	friday	0, 1	6	Fri Indicates if the event is to run on this day 0=No, 1=Yes
12	saturday	0, 1	6	Sat Indicates if the event is to run on this day 0=No, 1=Yes
13	record_id	1 to +2,147,483,647	11	Area Foreign key to id field in the area table. Record number of the area this event is for

Table 24. Area_event

COL #	Column name	Range value	Length	Meaning
14	area_online	0, 1, 2	6	Area Online Indicates if the area is to be put online or offline. 0=Do not care, 1=Online, 2=Offline
15	valid_routing	1 to +2,147,483,647	11	Valid Routing Foreign key to id field in the routing table. Record number of the type of routing to do for valid badge reads for an area
16	inv_routing	1 to +2,147,483,647	11	Invalid Routing Foreign key to id field in the routing table. Record number of the type of routing to do for invalid badge reads for an area
17	susp_routing	1 to +2,147,483,647	11	Suspended Routing Foreign key to id field in the routing table. Record number of the type of routing to do for suspended badge reads for an area
18	lost_routing	1 to +2,147,483,647	11	Lost Routing Foreign key to id field in the routing table. Record number of the type of routing to do for lost badge reads for an area
19	unk_routing	1 to +2,147,483,647	11	Unknown Routing Foreign key to id field in the routing table. Record number of the type of routing to do for unknown badge reads for an area
20	apb_routing	1 to +2,147,483,647	11	Antipassback Routing Foreign key to id field in the routing table. Record number of the type of routing to do for anti-passback badge reads for an area
21	escort_routing	1 to +2,147,483,647	11	Escort Routing Foreign key to id field in the routing table. Record number of the type of routing to do for escort badge reads for an area
22	reader_online	0, 1, 2	6	Reader Online Indicates if the reader is to be put online or offline. 0=Do not care, 1=Online, 2=Offline
23	phys_reader_type	0, 1, 2, 4, 8	6	Physical Reader Type 0=Do not care, 1=Badge only, 2=Badge and Keypad, 4=Keypad only, 8=Badge or Keypad
24	log_reader_type	0, 16, 32, 128, 256, 512, 2048	6	Logical Reader Type 0=Do not care, 16=APB In, 32=APB Out, 128=T&A In/Out, 256=T&A In, 512=T&A Out, 2048=Normal
25	swipe_reader_type	0, 1, 2	6	Swipe and Show Reader Type 0=Do not care, 1=Authorization Required, 2=No Authorization Required
26	nbadge	0, 1, 2	0	Num of Badges Indicates if it is a single or a double badge transaction reader. 0=Do not care, 1=Single, 2=Double
27	apb_duration	1 to +2, 147, 483, 647	11	Timed APB Duration Value represents number of seconds for timed APB.
28	door_lock	0, 1, 2	6	Door state 0=Do not care, 1=Unlocked, 2=Locked

Table 24. Area_event

COL #	Column name	Range value	Length	Meaning
29	open_too_long	0, 1, 2	6	Held Open Sensing 0=Do not care, 1=Detected, 2=Ignored
30	forced_open	0, 1, 2	6	Forced Open Monitoring 0=Do not care, 1=Detected, 2=Ignored
31	too_long_routing	1 to +2, 147, 483, 647	11	Held Open Routing Foreign key to id field in routing table. Record number for the type of routing to do for door held open alarm.
32	forced_routing	1 to +2, 147, 483, 647	11	Forced Open Routing Foreign key to id field in routing table. Record number for the type of routing to do for door forced open alarm.
33	two_man_rule	0 to 4	6	2MR Control Flag 0=Do not care, 1=Disabled, 2=Standard, 3=Modified Door Control, 4=Modified No Door Control
34	facility	1 to +2,147,483,647	11	Facility Foreign key to id field in facility table. Record number of the facility this record is in.
35	modify_date	>= 19700101	11	Modification date in the form yyyymmdd. System will supply the current date if no date is provided.
36	modify_time	0 to 235959	11	Modification time in the form hhmmss. System will supply the current time if no time is provided.
37 - 68	category1 - category32	1 to +2,147,483,647	11	Foreign keys to id's in category table. There are up to 32 category fields on an area event record.

Area_category

Table 25. Area_category

COL #	Column name	Range value	Length	Meaning
1	id*	1 to +2,147,483,647	11	Id Unique record number assigned by system except when using transaction type 35
2	area_id	1 to maximum record number in badge table	11	Area Id Foreign key to id field in area table. Record number of the area the category is in.
3	category_id	1 to maximum record number in category table	11	Category Id Foreign key to id field in category table. Record number of the category on the area with **`area_id`**.
4	slot_number	1 to 96	11	Slot Number Slot number the category is assigned to.
5	facility	1 to +2,147,483,647	11	Facility Foreign key to id field in facility table. Record number of the facility this record is in.
6	modify_date	>= 19700101	11	Modify Date Modification date in the form yyyymmdd. System will supply the current date if no date is provided.
7	modify_time	0 to 235959	11	Modify Time Modification time in the form hhmmss. System will supply the current time if no time is provided.

Area_event_category

Table 26. Area_event_category

COL #	Column name	Range value	Length	Meaning
1	id*	1 to +2,147,483,647	11	Id Unique record number assigned by system except when using transaction type 35
2	area_event_id	1 to maximum record number in badge table	11	Area Event Id Foreign key to **`id`** field in area_event table. Record number of the **`area_event`** the category is in.
3	category_id	1 to maximum record number in category table	11	Category Id Foreign key to **`id`** field in category table. Record number of the category on the **`area_event`** with **`area_event_id`**.
4	slot_number	1 to 96	11	Slot Number Slot number the category is assigned to.
5	facility	1 to +2,147,483,647	11	Facility Foreign key to id field in facility table. Record number of the facility this record is in.
6	modify_date	>= 19700101	11	Modify Date Modification date in the form yyyymmdd. System will supply the current date if no date is provided.
7	modify_time	0 to 235959	11	Modify Time Modification time in the form hhmmss. System will supply the current time if no time is provided.

Customer Survey

Title of this book: *Picture Perfect 4.0 Import/Export User Manual*

My overall rating of this book:

- ☐ Excellent
- ☐ Very good
- ☐ Good
- ☐ Satisfactory
- ☐ Fair
- ☐ Poor

What I liked most about this book:

What I would change, add, delete, etc. in future editions of this book:

Please return this form to:

Attn: Technical Writing
GE Security
791 Park of Commerce Blvd. Suite 100
Boca Raton, FL 33487

or e-mail your comments to:
bctdocs@ge.com

North America
T (561) 998-6100
T 888-GE-SECURITY
888-(437-3287)
F 561 998 6224
E rs-bctinfo@ge.com

Asia
T 852-2907-8108
F 852-2142-5063

Australia
T 61-3-9259-4700
F 61-3-9259-4799

Europe
T 32-2-725-11-20
F 32-2-721-86-13

Latin America
T 305-593-4301
F 305-593-4300

www.gesecurity.com

© 2005 General Electric Company
All Rights Reserved